HANDBOOK

FIREBIRD 6000

REMOTE CONTROL COMMANDS

Telecommunications
Techniques
Corporation

A DYNATECH COMPANY

... EXPECT EXCELLENCE

Digitized by the Internet Archive
in 2026

https://archive.org/details/manualsplus_12852

ML 10672
Rev. F

FIREBIRD 6000

REMOTE CONTROL COMMANDS HANDBOOK

FEBRUARY 1991

©Telecommunications Techniques Corporation

TELECOMMUNICATIONS TECHNIQUES CORPORATION
20410 Observation Drive
Germantown, Maryland 20876
(301) 353-1550 Fax. (301) 353-0731

TABLE OF CONTENTS

SECTION		PAGE
1.0	**INTRODUCTION**	**1-1**
2.0	**REMOTE CONTROL OPERATION**	**2-1**
2.1	INTRODUCTION	2-1
2.2	RS-232 PRINTER/CONTROLLER INTERFACE	2-1
2.3	IEEE-488 PRINTER/CONTROLLER OPERATION	2-8
3.0	**CONVENTIONS**	**3-1**
3.1	INTRODUCTION	3-1
3.2	MESSAGE EXCHANGE PROTOCOL	3-1
3.3	PROGRAM MESSAGES	3-2
3.4	RESPONSE MESSAGES	3-2
4.0	**REMOTE CONTROL COMMANDS**	**4-1**
4.1	INTRODUCTION	4-1
4.2	REMOTE CONTROL COMMANDS	4-2
5.0	**STATUS REPORTING**	**5-1**
5.1	INTRODUCTION	5-1
5.2	REGISTER GROUPS	5-1
5.3	QUEUE and BUFFER OPERATION	5-3
5.4	STATUS BYTE REGISTER GROUP	5-4
5.5	EVENT STATUS REGISTER GROUP	5-6
5.6	LINE SUMMARY REGISTER GROUP	5-8
5.7	INTERFACE STATUS REGISTER GROUP	5-11
6.0	**ERROR MESSAGES**	**6-1**
6.1	INTRODUCTION	6-1
6.2	ERROR MESSAGES	6-2
7.0	**IEEE-488.2 SPECIFICATION REQUIREMENTS**	**7-1**
7.1	INTRODUCTION	7-1
7.2	IMPLEMENTATION OF IEEE STANDARD 488.2-1987	7-1

FIGURE **PAGE**

5-1 FIREBERD 6000 Registers 5-1
5-2 Register Operation 5-3
5-3 Status Byte Register Group 5-4
5-4 Event Status Register Group 5-6
5-5 Line Summary Register Group 5-8
5-6 Interface Status Register 5-12

TABLE

2-1 RS-232 Pin Configuration 2-2
4-1 Standard Event Status Enable Register 4-5
4-2 Standard Event Status Register 4-7
4-3 Parallel Poll Enable Register 4-12
4-4 Service Request Enable Register 4-18
4-5 Status Byte Register 4-20
4-6 Interface Status Register 4-167
4-7 Interface History Status Register 4-169
4-8 Interface History Enable Status Register 4-171
4-9 Interface Loss Status Register 4-173
4-10 Interface Loss Enable Status Register 4-175
4-11 Line Status Register 4-177
4-12 Line History Status Register 4-179
4-13 Line History Enable Status Register 4-181
4-14 Line Loss Status Register 4-183
4-15 Line Loss Enable Status Register 4-185

SECTION 1
INTRODUCTION

The *FIREBERD 6000 Remote Control Handbook* provides information on controlling the FIREBERD 6000 from a remote device using either the RS-232 Printer/Controller or the IEEE-488 Printer/Controller Interface (Option 6005).

This handbook may be used cover-to-cover or as a reference to specific information. In addition, users should refer to their IEEE-488 documentation for information on programming their controller.

The *FIREBERD 6000 Remote Control Handbook* consists of the following sections:

- SECTION 2: OPERATION - describes the basic set-up and operation of the FIREBERD 6000 from a remote device.

- SECTION 3: REMOTE CONTROL CONVENTIONS - describes how to set up tests and read results when under remote control.

- SECTION 4: COMMANDS - presents all remote control commands alphabetically by group, describes each command's parameters, and provides examples.

- SECTION 5: STATUS REPORTING STRUCTURES - describes the registers and queues that can be used to monitor the FIREBERD 6000.

- SECTION 6: REMOTE CONTROL ERROR MESSAGES - contains error messages generated by the FIREBERD 6000.

- SECTION 7: IEEE-488.2 SPECIFICATION REQUIREMENTS - supplies information on how the FIREBERD 6000 has implemented the IEEE-488.2 standard.

SECTION 2
REMOTE CONTROL OPERATION

2.1 INTRODUCTION

This section provides information on controlling the FIREBERD 6000 from a remote device using either the RS-232 Printer/Controller Interface or the IEEE-488 Printer/Controller Interface (Option 6005). Remote control is possible using either a terminal or a programmed computer attached to the RS-232 Printer/Controller Interface. Remote control may also be accomplished by connecting the FIREBERD's IEEE-488 Printer/Controller Interface to an IEEE-488 bus.

2.2 RS-232 PRINTER/CONTROLLER INTERFACE

The RS-232 Printer/Controller Interface is located on the rear panel of the FIREBERD 6000. This interface allows connection by an asynchronous remote device. The interface connector is configured as a DCE and is an EIA RS-232-C-compatible, 25-pin, D-type female connector. Refer to Table 2-1 for pin assignments.

To communicate, the communication parameters of the FIREBERD and the remote controller must match. All communications through the RS-232 Printer/Controller Interface are asynchronous using standard ASCII characters. The interface can operate at baud rates of 110, 300, 600, 1200, 2400, 4800, or 9600 baud. The number of data bits can be either 7 or 8. The parity can be odd, even, or none. The baud rate, data bits, and parity are all set using the AUXILIARY category of the MENU switch.

2.2.1 RS-232 Remote Control Modes

When under RS-232 remote control, the FIREBERD 6000 can operate in either an interactive terminal mode or a non-interactive computer mode.

Table 2-1
RS-232 Pin Configuration

Pin No.	EIA Dgn.	Signal Description	Comments
1	AA	Protective Ground	Connected to chassis ground.
2	BA	Transmitted Data (TD)	Data transmitted to the FIREBERD.
3	BB	Received Data (RD)	Data received from the FIREBERD.
5	CB	Clear to Send (CTS)	Output HIGH when the FIREBERD is ready to receive.
6	CC	Data Set Ready (DSR)	Output HIGH when the FIREBERD is on.
7	AB	Signal Ground	Connected to circuit ground.
8	CF	Data Carrier Detect (DCD)	Output HIGH when the FIREBERD is on.
20	CD	Data Terminal Ready (DTR)	Input must be HIGH before FIREBERD can send data.

In the interactive terminal mode, the FIREBERD operates interactively with a remote device such as a dumb terminal or computer by: (1) providing a prompt character when the FIREBERD is ready to receive a command; (2) echoing all characters back to the remote device as they are typed; and (3) transmitting error messages when an execution or syntax error occurs.

In non-interactive computer mode, a programmed computer sends commands and receives predictable results from the FIREBERD. Thus, prompts, echo characters, and error messages are not sent in computer mode.

In both of these RS-232 remote control modes, the FIREBERD front-panel keys and switches are disabled (except for the ANALYSIS RESULTS switches and the ENTER key); the message **232 REMOTE CONTROL <ENTER> TO ABORT** appears in the display. If the front panel is placed in local lockout (see LOCKOUT command), **<ENTER> TO ABORT** is not displayed. This prevents the front panel from returning to local mode inadvertently.

2.2.2 RS-232 Terminal Mode

In terminal mode, the FIREBERD 6000 operates interactively with the remote device.

Initiating Terminal Mode

To set up the FIREBERD 6000 for terminal remote control (using the RS-232 Printer/Controller Interface), do the following:

1. Press the POWER switch to apply power to the FIREBERD.

2. Set the MENU switch to the AUXILIARY position.

3. Using the MENU keypad, type the number **11** and press the ENTER key to access Auxiliary Function 11 (I/O Driver Select). Press the SELECT softkey until RC:232 appears in the display.

4. Press the UP ARROW (▲) key.

5. Using the MENU keypad, type the number **38** and press ENTER to access the 232 auxiliary function.

6. Set the BAUD rate (110, 300, 600, 1200, 2400, 4800, or 9600), DATA bits (7 or 8), and PARITY (odd, even, or none) to match the terminal or terminal emulator's settings.

7. Cable the FIREBERD 6000.

8. From the terminal, send a <Control C>.

   RESULT: A prompt (>) appears on the terminal to indicate that the FIREBERD is ready to receive a command or query. If a prompt does not appear on the terminal, the interface may not be configured properly. Verify the set-up configuration and connections at both the FIREBERD and terminal, or attempt the auto-baud function from the remote device as described in Section 2.2.4.

When under remote control, the message **232 REMOTE CONTROL** appears in the FIREBERD's display.

Operating in Terminal Mode

In terminal mode, the prompt, echo, and error message functions are enabled. The prompt (>) indicates that the FIREBERD is ready to accept commands and is in interactive terminal mode. Wait for the prompt before sending additional commands to the FIREBERD. All characters are echoed on the terminal as they are typed. Commands may be edited using the backspace key and recalled by pressing the <ESC> key. Once the command has been correctly typed, press the ENTER key or RETURN key on your terminal to execute the command.

Each command must have the proper syntax and line terminator to be accepted by the FIREBERD. Valid line terminators are: LF (Line Feed), CR (Carriage Return), and CR LF (Carriage Return/Line Feed). When the FIREBERD receives a line terminator it executes all commands entered since the last terminator received. Each output generated in response to a command from the controller is terminated by a Carriage Return and Line Feed (<CR><LF>) and is sent to the terminal before the next prompt.

Ending Terminal Mode

To end terminal mode operation and return control of the FIREBERD to the front panel, send the LOCAL command *or* press the ENTER key.

2.2.3 RS-232 Computer Mode

In non-interactive computer mode, a programmed computer sends commands to the FIREBERD 6000 and, in turn, receives predictable results from the test instrument.

Initiating Computer Mode

To set up the FIREBERD 6000 for computer remote control (using the RS-232 Printer/Controller Interface), do the following:

1. Press the POWER switch to apply power to the FIREBERD.

2. Set the MENU switch to the AUXILIARY position.

3. Using the MENU keypad, type the number **11** and press the ENTER key to access Auxiliary Function 11 (I/O Driver Select). Press the SELECT softkey until RC:232 appears in the display.

4. Press the UP ARROW (▲) key.

5. Using the MENU keypad, type the number **38** and press ENTER to access the 232 auxiliary function.

6. Set the BAUD rate (110, 300, 600, 1200, 2400, 4800, or 9600), DATA bits (7 or 8), and PARITY (odd, even, or none) to match the terminal or terminal emulator's settings.

   NOTE: If the FIREBERD has not been placed in terminal mode since power has been applied, it is now in computer mode. The interface can also be placed in computer mode by sending the following command:

   RC_MODE COMPUTER

Operating in Computer Mode

Computer mode allows a programmed computer to control the FIREBERD. In computer mode, the prompt (>), echo, and command line editing are disabled; thus, the programmer can be sure that any data received by the computer is data that was specifically requested (via query).

Each command must have the proper syntax and line terminator to be accepted by the FIREBERD. The valid line terminator is LF (Line Feed - 0A Hex). If an invalid command is issued, the FIREBERD does not respond with an error message unless instructed to do so. (See the STATUS:ERROR? query described in Section 4.)

Ending Computer Mode

To end computer mode operation and control the FIREBERD from the front panel, send the LOCAL command *or* press the ENTER key.

2.2.4 RS-232 Auto-Baud Function

The FIREBERD 6000 features an auto-baud function that simplifies RS-232 remote control set-up. This function can be used to place the FIREBERD in terminal or computer mode, and allows a remote user to automatically configure the instrument's baud rate, data bits, and parity settings to those of the remote device. To initiate auto-baud from the remote control device, do the following:

1. Press the POWER switch to apply power to the FIREBERD.

2. Set the MENU switch to the AUXILIARY position.

3. Using the MENU keypad, type the number **11** and press the ENTER key to access Auxiliary Function 11 (I/O Driver Select). Press the SELECT softkey until RC:232 appears in the display.

4. Press your terminal's BREAK key twice at 1-second intervals.

   RESULT: The flashing message **AUTOBAUD IN PROGRESS** appears in the RESULTS display.

5. Repeatedly press your terminal's SPACE bar at a rate of 5 to 10 times per second for about 10 seconds or until the following message appears at the remote device.

   Auto-baud achieved. Press <ESCAPE> to continue.

6. Press the ESC key on the terminal. The FIREBERD responds as follows:

   Character format determined.

   RESULT: The FIREBERD is placed in the mode of choice prior to autobaud (TERMINAL or COMPUTER). To place the FIREBERD in the opposite mode, see the RC_MODE command in Section 4 of this manual. (Note that <Control C> can be used to place the FIREBERD in TERMINAL mode.)

   The following explains conditions that may occur in auto-baud mode.

- If auto-baud is not achieved within 30 seconds of initiation, the FIREBERD aborts the function and reverts to its previous RS-232 settings. This timeout is required because many terminals generate a BREAK signal as they power up or down, which causes the FIREBERD to enter auto-baud mode accidently. Under such conditions, the FIREBERD flashes the message **AUTO BAUD IN PROGRESS** in the RESULTS display until the 30-second timeout occurs. Note that the space bar also aborts auto-baud (if all instrument settings are correctly configured).

- The message **Auto-baud achieved. Press <ESCAPE> to continue.** is always transmitted with a format of 7 data bits, 2 stop bits, and even parity. For this reason, the message may appear garbled on terminals that are configured differently. Press your terminal's ESC key to determine the character format.

2.2.5 Installation Test for RS-232 Operation

To confirm that the FIREBERD 6000 has been properly configured and cabled for remote operation, do the following:

1. Power-up the remote computer or terminal.

2. Press the POWER switch to apply power to the FIREBERD.

3. Set the MENU switch to the AUXILIARY position.

4. Using the MENU keypad, type the number **11** and press the ENTER key to access Auxiliary Function 11 (I/O Driver Select). Verify that remote control is set to RS232.

5. Press the UP ARROW (▲) key.

6. Using the MENU keypad, type the number **38** and press ENTER to access the 232 auxiliary function. Verify that the computer interface parameters (baud, parity, and data bits) are set correctly.

7. Send the FIREBERD the following command with the proper terminator.

   *IDN?

8. Verify that the FIREBERD sends the following response:

   Telecommunications Techniques Corporation, FIREBERD 6000, 0, G

2.3 IEEE-488 PRINTER/CONTROLLER OPERATION

By connecting the FIREBERD 6000's IEEE-488 interface to an IEEE-488 bus, the instrument may be controlled by a programmed computer.

2.3.1 IEEE-488 Remote Control Mode

The FIREBERD 6000's IEEE-488 Printer/Controller Interface (Option 6005) complies with the IEEE Standard Digital Interface for Programmable Instrumentation (Std 488-1987). Located on the FIREBERD's rear panel, the IEEE-488 interface allows the test instrument to become part of an IEEE-488 bus system, thus enabling communication with a programmed controller.

The IEEE-488 interface operates in two modes: TALK-ONLY and ADDRESSABLE. When operating in TALK-ONLY mode, the FIREBERD should be connected to a listen-only printer. In ADDRESSABLE mode, the FIREBERD may be connected with up to 14 other devices, one of which must be a controller.

The DIP switch located on the IEEE-488 interface is used to select a mode and specify an address for the interface (in ADDRESSABLE mode only). The DIP switch configuration is as follows:

	SW1	SW2	SW3	SW4	SW5	SW6
(0=DOWN, 1=UP)	A0	A1	A2	A3	A4	Addressable Talk-Only

Where switch 6 selects the operating mode (ADDRESSABLE or TALK-ONLY) and switches 1-5 specify the unit address; switches 7 and 8 are not used. Note that remote control is only available in ADDRESSABLE mode.

For example, to set the FIREBERD to the ADDRESSABLE mode and specify address 28, do the following:

1. Set SW1 and SW2 to 0 (down).
2. Set SW3-SW6 to 1 (up).

2.3.2 Installing the IEEE-488 Remote Control Interface

The IEEE-488 interface is a user-installable option, as follows:

1. Make sure that power to the FIREBERD 6000 is turned OFF.

2. Insert the IEEE-488 card into the IEEE-488 connector in the FIREBERD's rear panel.

3. Set DIP switch 6 to ADDRESSABLE (i.e., the UP position).

4. Use DIP switches 1-5 to specify a unique address.

2.3.3 Setting Up the IEEE-488 Remote Control Interface

To set up the FIREBERD 6000 for remote control operation, do the following:

1. Press the POWER switch to apply power to the FIREBERD.

2. Set the MENU switch to the AUXILIARY position.

3. Using the MENU keypad, type the number **11** and press the ENTER key to access Auxiliary Function 11 (I/O Driver Select). Press the SELECT switch to set remote control to IEEE-488 (RC:488).

4. Press the ENTER key.

5. Make sure that DIP switch 6 on the IEEE-488 card is set to ADDRESSABLE and that a unique address has been specified using DIP switches 1-5.

6. Cable the FIREBERD 6000.

2.3.4 Initiating and Terminating Communication with the IEEE-488

To initialize and terminate communication with the IEEE-488 interface, refer to your controller manual and programming language reference manual.

2.3.5 Installation Test

To verify that the FIREBERD 6000 is configured and cabled correctly, do the following:

1. Apply power to the IEEE-488 controller.

2. Press the POWER switch to apply power to the FIREBERD.

3. Set the MENU switch to the AUXILIARY position.

4. Using the MENU keypad, type the number **11** and press the ENTER key to access Auxiliary Function 11 (I/O Driver Select). Verify that remote control is set to IEEE488 and that the correct address is specified using Auxilliary Function 39.

5. Send the FIREBERD the following command with the proper terminator:

   *IDN?

   NOTE: Refer to your controller documentation for details on how to address external devices and how to issue commands.

6. Verify that the FIREBERD sends the following response:

   Telecommunications Techniques Corporation, FIREBERD 6000, 0, G

2.3.6 If Test Fails

If the FIREBERD 6000 does not provide the specified results:

1. Check all cable connections.

2. Verify that the FIREBERD has been properly enabled and addressed.

3. Verify that the controller is configured properly using the documentation provided with the controller.

SECTION 3
CONVENTIONS

3.1 INTRODUCTION

This section provides information on the conventions used by the FIREBERD 6000 while under remote control. The communication between the FIREBERD and the controller is based on the IEEE-488.2 Specification, which standardizes the way instruments talk and listen to one another. By making these instruments consistent in behavior, control software becomes easier to write and maintain.

3.2 MESSAGE EXCHANGE PROTOCOL

The communication between the FIREBERD 6000 and the controller consists of messages known as commands, queries, and responses.

COMMANDS — Program the FIREBERD

Commands are sent from the controller to the FIREBERD. The FIREBERD interprets and executes each command issued. Commands can alter the FIREBERD set-up or affect the test in progress. Each command is composed of a header, associated data, and a terminator.

QUERIES — Obtain Set-Up Information and Results from the FIREBERD

Queries are sent from the controller to the FIREBERD. The FIREBERD interprets and responds to each query. Queries are composed of a header, immediately followed by a question mark (?).

RESPONSES — Answer a query

Responses are sent from the FIREBERD to the controller to answer a query.

3.3 PROGRAM MESSAGES

All communication sent from the controller to the FIREBERD 6000 is in the form of <program messages>. A <program message> is a series of commands and/or queries, each separated by a semicolon (;). A terminator is sent by the controller to identify the end of the <program message>. In RS-232 terminal mode operation, the terminator is either a CR (Carriage Return), LF (Line Feed), or CRLF (Carriage Return/Line Feed); in RS-232 computer mode operation, the terminator is LF; in IEEE-488 operation, the terminator can be LF or any character sent with an EOI (End Or Identify) signal asserted.

In RS-232 applications, the FIREBERD executes all commands and queries of the last program message in sequence when receiving a terminator from the controller. As commands and queries are processed and executed, space is made available in the input buffer. If the input buffer is full, the result is lost data, which generates errors.

In IEEE-488 applications, the FIREBERD executes commands and queries as it receives them. Commands and queries in the input buffer cannot be overwritten; thus, the FIREBERD stops accepting commands while the input buffer is full.

3.4 RESPONSE MESSAGES

The basic communication rule between the FIREBERD 6000 and the controller is that the FIREBERD only responds when the controller sends a query. In turn, the FIREBERD sends its response to the query before a new program message is sent.

In RS-232 applications, the FIREBERD sends a response as it is generated. In IEEE-488 applications, the response is stored in the output queue until the FIREBERD is addressed to talk. If a new program message is sent to the FIREBERD before the initial response, the initial response is discarded and a RESPONSE INTERRUPTED error is generated. The new program message is executed normally.

The communication sent from the FIREBERD 6000 to the controller is in the form of a <response message>. When using either the RS-232 terminal or computer mode of operation, the <response message> is sent immediately after the query is received. When using IEEE-488, the response message is sent when the FIREBERD is addressed to talk.

A <response message> consists of one <response message unit> per query of the preceding <program message>. If several responses are contained in a <response message> each is separated by a semicolon (;). The entire <response message> is ended by a terminator. The terminator for RS-232 terminal mode operation is CRLF; for RS-232 computer mode operation, LF; and for IEEE-488 operation, LF accompanied by an EOI signal. To determine the exact format of the <response message unit> for any given query, refer to Section 4.

SECTION 4
REMOTE CONTROL COMMANDS

4.1 INTRODUCTION

This section describes how to enter remote control commands with the proper command syntax, lists the commands by group, alphabetically describes each of the commands and parameters, and provides examples.

All examples shown in this section use the terminal mode format with printouts in a 40-column width. The commands are shown in upper-case characters. The commands may be entered in either upper- or lower-case characters.

Each remote control command is associated with a command group, as follows:

COMMON COMMANDS: Commands and queries that are common to all instruments.

AUXILIARY COMMANDS: Provide remote control access to the FIREBERD 6000's auxiliary functions.

CONFIGURATION COMMANDS: Remotely control any test configuration.

INTERFACE COMMANDS: Provide the ability to operate the FIREBERD 6000's interfaces via remote control.

JITTER COMMANDS: Provide remote control access to the FIREBERD 6000's jitter generation and measurement capability.

PRINT COMMANDS: Control the FIREBERD 6000's print capability.

REMOTE CONTROL COMMANDS: Set functions that are not grouped with other commands.

RESULT COMMANDS: Retrieve test results.

STATUS COMMANDS: Set and read the status reporting registers, queues, and buffers.

4.2 REMOTE CONTROL COMMANDS

This section presents all of the remote control commands in their associated group in alphabetical order. Each command is listed on a separate page; for ease of location, the command name appears in the upper-left corner and the command type appears in the upper-right corner of each page. Each page includes an explanation of the command and a brief example of command usage. Bold characters are used throughout this section to indicate the commands as well as the variables to be entered with the commands.

COMMON COMMANDS

The IEEE-488.2 Standard defines a set of commands and queries that should be common to all instruments. Some commands are mandatory, while others are optional.

The common commands and queries accepted by the FIREBERD 6000 are:

Commands	Queries
*CLS	
*ESE	*ESE?
	*ESR?
	*IDN?
	*IST?
*OPC	*OPC?
	*OPT?
*PRE	*PRE?
*PSC	*PSC?
*RCL	
*RST	
*SAV	
*SRE	*SRE?
	*STB?
*TRG	
	*TST?
*WAI	

*CLS

Clear Status **command**

***CLS** :clear the Standard Event Status Register

The ***CLS** command clears the Standard Event Status Register, including the device-defined print queue and the error buffer. The *CLS command does not clear the output queue if it is part of a <program message> containing a query. This means the Message Available (MAV) bit of the Status Register may not be cleared. (See Section 5 for additional information.)

Example:

```
>*CLS          :clear the status register, queues, and buffers
>*ESR?         :read the register
 0             :status registers cleared
```

*ESE

Event Status Enable **command/query**

***ESE** :set the bits in the Standard Event Status Enable Register

***ESE?** :return the status of the bits in the Standard Event Status Enable Register

The ***ESE** command sets the bits in the Standard Event Status Enable Register. The value of the bits determine which bits in the Standard Event Status Register are enabled. When a bit in the Standard Event Status Register and its corresponding enable bit are set, the Event Status Bit 5 (ESB) of the Status Register is also set. (See the *ESR and *STB commands, and Section 5 for additional information.)

The ***ESE?** query returns the status of the bits in the Standard Event Status Enable Register. The value returned is the total bit weight of all the bits that are enabled when the byte is read. Table 4-1 offers information on the bits in the Standard Event Status Enable Register.

Table 4-1
Standard Event Status Enable Register

Bit	Weight	Enables:
7	128	PON - Power On
6	64	URQ - User Request
5	32	CME - Command Error
4	16	EXE - Execution Error
3	8	DDE - Device Dependent Error
2	4	QYE - Query Error
1	2	RQC - Request Control
0	1	OPC - Operation Complete

*ESE (Continued)

Event Status Enable **command/query**

Example:

```
>*ESE 32          :enable bit 5 (Command Error) of the register
>*ESE?            :report the enabled bit(s) in the register
32
```

*ESR

Event Status Register — query

***ESR?** :return the current contents of the Standard Event Status Register

The ***ESR?** query returns the contents of the Standard Event Status Register. The value returned is the total weight of all the bits that are enabled when the byte is read. After reading the register, this query is cleared. Table 4-2 offers information on each bit in this register. (See the *SRE and *OPC commands and Section 5 for additional information.)

Table 4-2
Standard Event Status Register

Bit	Weight	Bit Name	Condition
7	128	PON	0 = Not powered-on since last reading. 1 = Powered-on since last reading.
6	64	URQ	Not defined. Always zero.
5	32	CME	Set to one when a Command error occurs.
4	16	EXE	Set to one when an Execution error occurs.
3	8	DDE	Not defined. Always zero.
2	4	QYE	Set to one when a Query error occurs.
1	2	RQC	Request control - not used - always zero.
0	1	OPC	Set to one when a program message containing the *OPC command is executed.

Example:

```
>*ESR?                    :read the Event Status Register
32                        :a Command error occurred
```

*IDN

Identification Query — query

***IDN?** :return FIREBERD 6000 identification information

The ***IDN?** query returns the FIREBERD 6000's identification information, as follows: manufacturer, model, serial number (always 0), and software revision.

The ***IDN?** query must be the last query in a program. Queries following *IDN? in a program message are ignored.

Example:

>*IDN? :return the FIREBERD's identification information

Telecommunications Techniques Corporation, FIREBERD 6000, 0, G

*IST

Individual Status **query**

***IST?** :return the Individual Status message (IST)

The ***IST?** query returns the Individual Status message (IST), which determines how the FIREBERD 6000 responds to a parallel poll.

When the controller issues a parallel poll, it sends a sense bit. The FIREBERD respond to the parallel poll only if the IST (Individual Status) value matches the sense of the serial poll. The value of the IST message is determined by the Status Byte Register and the Parallel Poll Enable Register. (See the *PRE and *STB commmands, and Section 5 for additional information.)

Example:

>***IST?** :return the Individual Status message
0

*OPC

Operation Complete **command/query**

***OPC** :set the Operation Complete (OPC) bit of the Standard Event Status Register

***OPC?** :return the status of the OPC bit

The ***OPC** command sets the Operation Complete (OPC) bit of the Standard Event Status Register to a value of 1 when all pending device operations are completed.

The ***OPC?** query places an ASCII "1" in the output queue when all pending device operations are completed (i.e., no operations are pending).

Example:

>CONFIG:SELF_LOOP ON;*OPC :set the OPC bit to a value of 1 after self-loop mode has been established

>CONFIG:SELF_LOOP ON;*OPC? :returns a 1 after self-loop mode has been established

1

*OPT

Option Identification — query

***OPT?** :return a list of installed options

The ***OPT?** query reports on the options that are installed in the FIREBERD 6000. Options are reported by field, as follows:

Field 1: G.703 JIT (G.703 Jitter)
Field 2: JIT SA (Jitter Spectrum Analysis)
Field 3: DS1 WIDEBAND JIT (DS1 Wideband Jitter)
Field 4: DS1 JIT GEN (DS1 Jitter Generator)
Field 5: CLK RECOV (Clock Recovery)
Field 6: IEEE488 (IEEE-488 Interface)
Field 7: REMOTE (Remote Control FIREBERD)
Field 8: SYNC USER MSG (Synchronous User Message)
Field 9: Reserved for future use
Field 10: Reserved for future use
Field 11: Reserved for future use

When an option is not installed, a zero appears in lieu of the option name.

Example:

>OPT? :identify the options that are installed in the FIREBERD

0,0,0,0,0,IEEE488,0,SYNC USER MSG,0,0,0

*PRE

Parallel Poll Enable **command/query**

***PRE** :set the bits in the Parallel Poll Enable Register

***PRE?** :return the status of the bits in the Parallel Poll Enable Register

The ***PRE** command sets the bits in the Parallel Poll Enable Register. The value of these bits determine which bits of the Status Byte will set the Individual Status (IST) message. The IST message is enabled whenever a bit in the Parallel Poll Enable Register and the corresponding bit in the Status Byte Register are set. The IST message is used to determine the response to a parallel poll. (See the *STB and *IST commands, and Section 5 for more information.)

The ***PRE?** query returns the current status of the enable register. The value returned is the total weight of all the bits that are enabled when the byte is read. Table 4-3 offers information on each bit in the Parallel Poll Enable Register.

Table 4-3
Parallel Poll Enable Register

Bit	Weight	Enables this bit to set IST:
7	128	Not Defined.
6	64	MSS - Master Status Summary.
5	32	ESB - Event Status Bit.
4	16	MAV - Message Available.
3	8	IEB - Interface Event Bit.
2	4	LEB - Line Event Bit.
1	2	PAV - Print Available.
0	1	EAV - Error Message Available.

*PRE (Continued)

Parallel Poll Enable **command/query**

Example:

```
>*PRE 16          :enable MAV to set IST
>*PRE?            :read current value of the Parallel Poll Enable
                   Register
16
```

*PSC

Power-On Status Clear **command/query**

***PSC** :clear all Status Enable Registers at FIREBERD 6000 power-on

***PSC?** :return the current status of the activity

The ***PSC** command sets the FIREBERD 6000's status clear bit. When set to 1, the FIREBERD automatically clears all Status Enable Registers at power-on, which prevents the instrument from generating a service request; when set to 0, the FIREBERD does not clear the Status Enable Registers.

The ***PSC?** query returns the current status of the activity.

Example:

>***PSC 0** :preserve all Status Enable Register settings at power-on

>***PSC?** :return the current setting of the bit

0

*RCL

Recall Stored Test Set-ups — command

***RCL** :recall a stored test set-up

The ***RCL** command recalls a stored test set-up for immediate use by the FIREBERD 6000. This command is identical to the RECALL function of the instrument's MENU switch. Note, however, that the remote control and printer settings specified in Auxiliary Function 11 are not changed; neither are the RS-232 settings in Auxiliary Functions 38 or the Status Clear bit (see the *PSC command.) This ensures that remote control operation can continue after the *RCL command is issued. Valid program numbers are 0-9.

Example:

>*RCL 3 :recall stored test set-up number 3

*RST

Reset **command**

***RST** :reset the FIREBERD 6000 to its factory-default settings

The ***RST** command sets the FIREBERD 6000 to its factory-default settings.

Example:

>***RST** :set the FIREBERD 6000 to its factory-default settings

*SAV

Save **command**

***SAV n** :save a stored test set-up

The ***SAV** command saves a test set-up (n = 0-9). This command is similar in function to the STORE activity on the FIREBERD 6000's MENU switch. Valid program numbers are 0-9.

>***SAV 3** :save current configuration as test set-up number 3

*SRE

Service Request Enable **command/query**

***SRE** :set the bits in the Service Request Enable Register

***SRE?** :return the status of the bits in the Service Request Enable Register

The ***SRE** command sets the bits in the Service Request Enable Register. The value of these bits determine which bits in the Service Request Register are enabled. When a bit in the Service Request Register and its corresponding enable bit are set, the Master Summary Status (MSS) bit of the Status Register is also set. The MSS bit represents the summary of all the status registers, queues and buffers. When set to 1, a service request (SRQ) is signaled by the FIREBERD 6000 when operating under IEEE-488 remote control. (See the *STB command and Section 5 for more information.)

The ***SRE?** query returns the status of the bits in the Service Request Enable Register. The value returned is the total weight of all the enabled bits at the time the byte is read. Table 4-4 offers information on the bits in the Service Request Enable Register.

Table 4-4
Service Request Enable Register

Bit	Weight	Enables:
7	128	Not defined. Always zero.
6	64	MSS - Master Status Summary.
5	32	ESB - Event Status Bit.
4	16	MAV - Message Available.
3	8	IEB - Interface Event Bit.
2	4	LEB - Line Event Bit.
1	2	PAV - Print Available.
0	1	EAV - Error Message Available.

***SRE (Continued)**

Service Request Enable **command/query**

Example:

>*SRE 16	:bit 4 MAV (Message Available) sets the MSS bit
>*SRE?	:read the SRE register
16	:FIREBERD responds with the current value of the SRE register

*STB

Status Byte **query**

***STB?** :return the current value of the FIREBERD 6000's status byte

The ***STB** query returns the current value of the FIREBERD 6000's status byte. The value returned is the total weight of all the bits that are enabled when the byte is read. The Status Byte Register contains the same value that is read by a serial poll over the IEEE-488 bus. Table 4-5 offers information on each bit in the Status Byte Register. (See the *SRE command and Section 5 for more information.)

Table 4-5
Status Byte Register

Bit	Weight	Enables:
7	128	Not defined. Always zero.
6	64	MSS - Master Status Summary.
5	32	ESB - Event Status Bit.
4	16	MAV - Message Available.
3	8	IEB - Interface Event Bit.
2	4	LEB - Line Event Bit.
1	2	PAV - Print Available.
0	1	EAV - Error Message Available.

Example:

>*STB? :read the FIREBERD's Status Byte Register
33 :33 indicates that the ESB and EAV bits are set

*TRG

Trigger **command**

***TRG** :clear all results

The ***TRG** command executes a device trigger which clear all results. This command is identical to the **RESULT:CLEAR** command, and performs the same function as a device trigger over the IEEE-488 bus.

Example:

>***TRG** :FIREBERD 6000 clears results

*TST

Self-Test **query**

***TST?** :report self-test results

The ***TST** query indicates whether the FIREBERD 6000 passed or failed its last self-test. If the *TST query returns a value of 0, no problems were encountered; if a value of 1 is returned, it indicates that problems occurred during the self-test (e.g., ROM failure, RAM failure, NOVRAM loss, etc.).

Example:

```
>*TST?          :report on the last self-test
0               :self-test was successful
```

*WAI

Wait **command**

***WAI** :delay execution of any commands until all operations are complete

The ***WAI** command causes all previous commands to complete before executing any other commands or queries.

NOTE: Since the FIREBERD executes all commands serially, the ***WAI** command does not need to be used. However, because it is a required IEEE-488-standard command, it is included in the Common Command set.

Example:

>MODE 2M;*WAI :enables all commands prior to *WAI to be executed prior to completing those that were issued after this command was invoked

AUXILIARY COMMANDS

The FIREBERD 6000 features a set of commands and queries that offer remote access to the instrument's auxiliary functions.

The auxiliary commands and queries accepted by the FIREBERD 6000 are:

Commands	Queries
AUX:BLOCK_LEN	AUX:BLOCK_LEN?
AUX:CLEAR_MEM	
AUX:COMMON_FREQS	AUX:COMMON_FREQS?
AUX:DATE	AUX:DATE?
	AUX:DELAY?
AUX:DELAY:START	AUX:DELAY:START?
AUX:DELAY:STOP	AUX:DELAY:STOP?
	AUX:DEVICE_SEL?
AUX:GEN_CLK_POL	AUX:GEN_CLK_POL?
AUX:GEN_DATA_INV	AUX:GEN_DATA_INV?
AUX:IN_USE:CLEAR	
	AUX:IEEE488:ADDR?
AUX:INBAND_FLOW	AUX:INBAND_FLOW?
	AUX:IN_USE?
AUX:OUTBAND_FLOW	AUX:OUTBAND_FLOW?
AUX:PATTERN:APPEND	
AUX:PATTERN:INSERT	
AUX:PATTERN:REPLACE	
AUX:PATTERN:DELETE	
AUX:PATTERN:DATA	AUX:PATTERN:DATA?
	AUX:PATTERN:SIZE?
AUX:PATTERN	AUX:PATTERN?
AUX:PRGM_PATTERN	AUX:PRGM_PATTERN?
AUX:RCV_CLK_POL	AUX:RCV_CLK_POL?
AUX:RCV_CLK_SEL	AUX:RCV_CLK_SEL?
AUX:RS232?	
AUX:RS232:BAUD	AUX:RS232:BAUD?
AUX:RS232:DATA_BITS	AUX:RS232:DATA_BITS?
AUX:RS232:PARITY	AUX:RS232:PARITY?
AUX:SINGLE_XMIT	AUX:SINGLE_XMIT?
AUX:SITE_ID	AUX:SITE_ID?
AUX:SYNCLOSS_ACT	AUX:SYNCLOSS_ACT?
AUX:SYNCLOSS_THR	AUX:SYNCLOSS_THR?
AUX:TIME	AUX:TIME?
AUX:USR_SYNC_THR	AUX:USR_SYNC_THR?
AUX:USR_SYNC_THR:BYTES	AUX:USR SYNC_THR:BYTES?
AUX:ASYN_TIMEOUT	AUX:ASYN:TIMEOUT?

ASYN_TIMEOUT

Asynchronous Data Timeout **command/query**

AUX:ASYN_TIMEOUT [OFF|ON] :set the timeout status for asynchronous data

ASYN_TIMEOUT? :report timeout status for asynchronous data

The **AUX:ASYN_TIMEOUT** command sets the timeout status for asynchronous data. ON forces a synchronization loss if no data is received within a time period of 10 seconds; OFF does not force a timeout.

The **AUX:ASYN_TIMEOUT?** query reports the current status of the timeout function for asynchronous data.

This remote control command is identical to **Auxiliary Function 46** when the FIREBERD is controlled from the front panel.

Example:

>AUX:ASYN_TIMEOUT? :report the current status of asynchronous data timeout

OFF :the asynchronous timeout function is off

>AUX:ASYN_TIMEOUT ON :activate the asynchronous timeout function

BLOCK_LEN

Block Length **command/query**

AUX:BLOCK_LEN (block length) :set block length

AUX:BLOCK_LEN? :report the current block length

The **AUX:BLOCK_LEN** command sets the number of bits in each block (as referred to by the block analysis results). The (block length) can be any number of bits within the range 100 to 1000000.

This remote control command is identical in function to **Auxiliary Function No. 30** when the FIREBERD is controlled from the front panel.

Example:

>AUX:BLOCK_LEN? :report the current block length
1000 bits
>AUX:BLOCK_LEN 25000 :make the new block length 25,000 bits

CLEAR_MEM

Clear NOVRAM **command**

AUX:CLEAR_MEM :clear NOVRAM

The **AUX:CLEAR_MEM** command clears all of the FIREBERD 6000's non-volatile RAM (NOVRAM) in the FIREBERD.

This command should be used with extreme caution since it wipes out all preset configurations, auxiliary functions, front-panel programs, etc.

This remote control command is identical in function to **Auxiliary Function No. 43** when the FIREBERD is controlled from the front panel.

Example:

>AUX:CLEAR_MEM :clear NOVRAM

COMMON_FREQS

Program Common Frequencies **command/query**

AUX:COMMON_FREQS (number), (frequency) :set number and frequency

AUX:COMMON_FREQS? (number) :report the current synthesizer frequency assigned to a specific number

The **AUX:COMMON_FREQS** command allows the user to program commonly used synthesizer frequencies by number. The (number) can be any integer within the range 1-11; (frequency) can be any frequency within the range .05-15000 kHz. The default frequencies are identical to those on the FIREBERD 6000's front panel.

The **AUX:COMMON_FREQS?** query reports the current frequency assigned to the (number) specified.

This remote control command is identical in function to **Auxiliary Function No. 32** when the FIREBERD is controlled from the front panel.

Example:

>AUX:COMMON_FREQS? 2 :report the current synthesizer frequency assigned to No. 2

>AUX:COMMON_FREQS 2,2048 :change synthesizer frequency No. 2 to 2048 kHz

DATE

Set and Display Date **command/query**

AUX:DATE "(date)" :set the date

AUX:DATE? :report the date

The **AUX:DATE** command sets the current date. The date is entered within quotes by month, day, and year.

The **AUX:DATE?** query reports the currently set date.

This remote control command is identical in function to **Auxiliary Function No. 60** when the FIREBERD is controlled from the front panel.

Example:

>AUX:DATE "10/14/91" :set the date to Oct. 14, 1991
>AUX:DATE? :report the current date
"10/14/91"

DELAY

Delay Measurement **command/query**

AUX:DELAY:START "(event)" :set the start event for the delay measurement

AUX:DELAY:STOP "(event)" :set the stop event for the delay measurement

AUX:DELAY:START? :report the current start event

AUX:DELAY:STOP? :report the current stop event

AUX:DELAY? :report the current start and stop events

This series of commands and queries define the parameters for the FIREBERD 6000's delay measurement.

The **AUX:DELAY:START** and **AUX:DELAY:STOP** commands set the start and stop events. Delay measurements start upon detection of the START "(event)" and stop on the detection of the STOP "(event)". The valid START and STOP "(event)"s are:

DTR/	:rising edge of interface DTR signal
DTR	:falling edge of interface DTR signal
DSR/	:rising edge of interface DSR signal
DSR	:falling edge of interface DSR signal
RTS/	:rising edge of interface RTS signal
RTS	:falling edge of interface RTS signal
CTS/	:rising edge of interface CTS signal
CTS	:falling edge of interface CTS signal
RLS/	:rising edge of interface RLSD signal
RLS	:falling edge of interface RLSD signal
TP/	:rising edge of rear panel test point
TP	:falling edge of rear panel test point
GPA or **RPA**	:generator or receiver pattern (START or STOP respectively)

DELAY (Continued)

Delay Measurement — command/query

GDA/ or **RDA/** :generator or receiver data (START or STOP) rising edge

GDA or **RDA** :generator or receiver data (START or STOP) falling edge

The **AUX:DELAY?** query reports the current start and stop events. The **AUX:DELAY:START?** and **AUX:DELAY:STOP?** individually report the current start and stop events, respectively.

This remote control command is identical in function to **Auxiliary Function No. 31** when the FIREBERD is controlled from the front panel.

Example:

>AUX:DELAY:START? :report the current delay start event

"RTS/" :the rising edge of the RTS interface signal is the current delay start event

>AUX:DELAY:START "CTS/" :change the delay start event to the rising edge of the CTS interface signal

>AUX:DELAY:STOP? :report the current delay stop event

"DTR\" :the falling edge of the DTR interface signal is the current delay stop event

>AUX:DELAY:STOP "DTR/" :change the delay stop event to the rising edge of the DTR interface signal

>AUX:DELAY? :report the current delay start and stop events

"RTS/, DTR"

DEVICE_SEL

Report Current Remote Control and Printer Settings query

AUX:DEVICE_SEL? :report the currently selected standard for remote control and print devices

The **AUX:DEVICE_SEL?** query reports the currently selected standard for remote control and print devices (RS-232 or IEEE-488). When issuing the query, the remote control setting is displayed first, followed by the printer setting.

This remote control command is identical in function to **Auxiliary Function No. 11** when the FIREBERD is controlled from the front panel.

Example:

>**AUX:DEVICE_SEL?** :report the currently selected standard for remote control and printing

488, 232

GEN_CLK_POL

Generator Clock Polarity **command/query**

AUX:GEN_CLK_POL [NORMAL|INVERT] :set generator clock polarity

AUX:GEN_CLK_POL? :report the current status of generator clock polarity

The **AUX:GEN_CLK_POL** command sets the FIREBERD's generator clock polarity. NORMAL sets generator clock polarity to the current interface standard; INVERT sets generator clock polarity to the inverse of the interface standard. NORMAL is the default.

The **AUX:GEN_CLK_POL?** query reports the current status of generator clock polarity.

This remote control command is identical in function to **Auxiliary Function No. 1** when the FIREBERD is controlled from the front panel.

Example:

>AUX:GEN_CLK_POL NORMAL :set generator clock polarity to normal

>AUX:GEN_CLK_POL? :then display its status
NORMAL

GEN_DATA_INV

Generator Data Invert **command/query**

AUX:GEN_DATA_INV [OFF|ON] :select inverted/non-inverted data generation

AUX:GEN_DATA_INV? :report the current status

The **AUX:GEN_DATA_INV** command determines whether data generated by the FIREBERD is normal or inverted. OFF selects non-inverted data; ON inverts all data generated by the instrument. OFF is the default selection.

The **AUX:GEN_DATA_INV?** query reports the current status of this activity.

This remote control command is identical in function to **Auxiliary Function No. 6** when the FIREBERD is controlled from the front panel.

Example:

>AUX:GEN_DATA_INV? :display the current status
ON :all generated data is currently inverted
>AUX:GEN_DATA_INV OFF :turn inverted data off

IEE488:ADDR

Return the Current IEEE-488 Address — query

AUX:IEEE488:ADDR? :report the current IEEE-488 address

The **AUX:IEEE488:ADDR?** query reports the current IEEE-488 address.

This remote control command is identical in function to **Auxiliary Function No. 39** when the FIREBERD is controlled from the front panel.

Example:

>AUX:IEEE488:ADDR? :report the current IEEE-488 address
20

INBAND_FLOW

In-Band Flow Control **command/query**

AUX:INBAND_FLOW [OFF|ON] :enable or disable in-band flow control

AUX:INBAND_FLOW? :report the current status of in-band flow control (on or off)

AUX:INBAND_FLOW:[XON(#Hnn) |XOFF(#Hnn)] :specify the XON or XOFF characters for in-band flow control in hexadecimal format

AUX:INBAND_FLOW:[XON|XOFF]? :report the current XON or XOFF characters

This series of commands and queries supports in-band flow control.

The **AUX:INBAND_FLOW** command turns in-band flow control off or on, respectively. When OFF is selected, the FIREBERD does not respond to XON and XOFF characters in the received data.

The **AUX:INBAND_FLOW?** query reports the current status of in-band flow control (i.e., on or off).

The **AUX:INBAND_FLOW:[XON|XOFF]** command designates the XON or XOFF characters used in in-band flow control. #H indicates that the code immediately following is in the hexadecimal format; nn is the actual hexadecimal code (range: 00-FF).

The **AUX:INBAND_FLOW:[XON|XOFF]?** query reports the currently designated XON or XOFF characters.

This remote control command is identical in function to **Auxiliary Function No. 9** when the FIREBERD is controlled from the front panel.

INBAND_FLOW (Continued)

In-Band Flow Control **command/query**

Example:

>AUX:INBAND_FLOW?	:report the current state of in-band flow control
OFF	:in-band flow control is off
>AUX:INBAND_FLOW ON	:turn in-band flow control on
>AUX:INBAND_FLOW XON?	:report the current XON character
#H0D	:0D is the current XON character
>AUX:INBAND_FLOW:XON #HFF	:change the XON character to FF
>AUX:INBAND_FLOW XOFF?	:report the current XOFF character
>#HFF	:FF is the current XOFF character
>AUX:INBAND_FLOW:XOFF #H0A	:change the XOFF character to 0A

IN_USE

Auxiliary Functions In Use **command/query**

AUX:IN_USE? :report the current auxiliary functions in use

AUX:IN_USE:CLEAR :set the auxiliary functions to their defaults

The **AUX:IN_USE?** query reports whether auxiliary functions 1-10 are in use or in their default states. This remote control command features 9 fields, one for each active auxiliary function as follows:

Field 1: GEN_CLK_POL
Field 2: RCV_CLK_POL
Field 3: ACT_SYN_LOS
Field 4: SINGLE_XMIT
Field 5: GEN_DAT_INV
Field 6: RCV_CKL_SEL
Field 7: OUT_BND_FLW
Field 8: IN_BND_FLW
Field 9: SYN_LOS_THR

NOTE: Auxiliary Function 5 is not currently in use.

When in the default states, a zero appears in the field in lieu of the auxiliary function name.

The **AUX:IN_USE:CLEAR** commands returns all auxiliary functions to their default states.

This remote control command is identical in function to **Auxiliary Function No. 99** when the FIREBERD is controlled from the front panel.

Example:

>AUX:IN_USE? :report the current auxiliary functions in use

0,RCV_CLK_POL,0,0,0,0,
OUT_BAND_FLW,0,0 :receiver clock polarity and out-of-band flow control are in use

>AUX:IN_USE:CLEAR :set the auxiliary functions to their defaults

OUTBAND_FLOW

Out-of-Band Flow Control **command/query**

AUX:OUTBAND_FLOW [OFF|ON] :enable or disable out-of-band flow control

AUX:OUTBAND_FLOW? :report the current status of out-of-band flow control (on or off)

AUX:OUTBAND_FLOW:
[TR|DM|RS|CS|RR]
[OFF|ON] :turn individual signaling leads on or off

AUX:OUTBAND_FLOW:
[TR|DM|RS|CS|RR]? :report the status of the individual signaling leads

Out-of-band flow control is a function which allows data transmission to take place only when certain signaling leads are placed in the ON state (i.e., enabled).

The **AUX:OUTBAND_FLOW** command turns out-of-band flow control off or on, respectively.

The **AUX:OUTBAND_FLOW:[TR|DM|RS|CS|RR]** command enables or disables the signaling lead(s) that will control data transmission. These signaling leads can be enabled even when out-of-band flow control is OFF.

The **AUX:OUTBAND_FLOW?** query reports the current status of out-of-band flow control (i.e., on or off).

The **AUX:OUTBAND_FLOW:[TR|DM|RS|CS|RR]?** query reports the status of the individual signaling leads (i.e., on or off).

This remote control command is identical in function to **Auxiliary Function No. 8** when the FIREBERD is controlled from the front panel.

OUTBAND_FLOW (Continued)

Out-of-Band Flow Control — command/query

Note that RS-449-standard signal names are used in this command. The relationship between RS-449 and RS-232 signal names is as follows:

RS-449	RS-232
TR	DTR
DM	DSR
RS	RTS
CS	CTS
RR	RLSD

Example:

```
>AUX:OUTBAND_FLOW?          :report the current state of out-of-
                             band flow control
OFF                         :out-of-band flow control is off
>AUX:OUTBAND_FLOW ON        :turn out-of-band flow control on
>AUX:OUTBAND_FLOW:TR?       :report the current status of the
                             TR signaling lead
>OFF                        :the TR signaling lead is off
>AUX:OUTBAND_FLOW:TR ON     :turn the TR signaling lead on
>AUX:OUTBAND_FLOW:CS?       :report the current status of the
                             CS signaling lead
>ON                         :the CS signaling lead is on
>AUX:OUTBAND_FLOW:CS OFF    :turn the CS signaling lead off
>AUX:OUTBAND_FLOW:DM?       :report the current status of the
                             DM signaling lead
>OFF                        :the DM signaling lead is off
>AUX:OUTBAND_FLOW:DM ON     :turn the DM signaling lead on
```

PATTERN

User Message Selection **command/query**

AUX:PATTERN [USER1|USER2|USER3|
1:7|3IN24|T1-1|T1-2|T1-3|T1-4|
|T1-5|T1-6|T1-MW|DDS-1|DDS-2|
|DDS-3|DDS-4|DDS-5|DDS-6] :select a data pattern

AUX:PATTERN:DATA
[USER1|USER2|USER3],
"(string)" :create a data pattern and assign it to the USER message specified

AUX:PATTERN:APPEND
[USER1|USER2|USER3],
"(string)" :append string to the specified USER message

AUX:PATTERN:INSERT
[USER1|USER2|USER3],
(position),"(string)" :insert string into USER message beginning at the byte position specified

AUX:PATTERN:REPLACE
[USER1|USER2|USER3],
(position),"(string)" :replace string in the USER message beginning at the byte position specified

AUX:PATTERN:DELETE
[USER1|USER2|USER3],
(position),(bytes) :delete a number of bytes in the string of the USER message beginning at the byte position specified

AUX:PATTERN? :report the currently selected pattern

AUX:PATTERN:SIZE?
[USER1|USER2|USER3] :report the length of the user pattern specified

PATTERN (Continued)

User Message Selection **command/query**

AUX:PATTERN:DATA?
[USER1|USER2|USER3] :report the current pattern assigned to the USER message specified

This series of commands and queries is dedicated to the creation and selection of user patterns and messages.

The **AUX:PATTERN** command is used to select from a series of "canned" patterns which are supplemental to those available through the DATA switch: 1:7, 3-in-24, T1-1/T1-6, T1-mW, and DDS-1/6. This command is also used to select one of three user-created patterns of up to 2048 characters each: USER1-USER3.

NOTE: When entering a lengthy user-created pattern, the number of characters that can be entered on any given line is limited in RS-232 terminal mode. If the pattern to be entered is longer than a single line, hit the RETURN or ENTER key before the bytes entered exceed line space; in such instances, a new prompt appears on the line immediately following and you may continue entering bytes until pattern entry is completed. *Note that the entire pattern must always be placed between quotation marks.*

The **AUX:PATTERN:DATA** is used to create a data pattern and assign it to the USER message specified. Spaces must be inserted between each byte in the data pattern.

As their names imply, the **AUX:PATTERN:APPEND**, **AUX:PATTERN:INSERT**, **AUX:PATTERN:REPLACE**, and **AUX:PATTERN:DELETE** commands are used to modify the user-created pattern specified.

The **AUX:PATTERN?** query reports the currently selected pattern (via the AUX:PATTERN command).

The **AUX:PATTERN:SIZE?** query reports the length of the user-created pattern specified.

PATTERN (Continued)

User Message Selection — command/query

The **AUX:PATTERN:DATA?** query reports the current user-created pattern assigned to the USER message specified.

This remote control command is identical in function to **Auxiliary Function No. 41** when the FIREBERD is controlled from the front panel.

Example:

>AUX:PATTERN:DATA? USER1 :report the current user-created pattern assigned to USER1

"48 45 4C 4C 4F" :the USER1 pattern is HELLO in ASCII

>AUX:PATTERN:DATA USER2 "48 49" :set USER2 pattern to HI in ASCII

>AUX PATTERN:DATA? USER2 :report the current user-created pattern assigned to USER2

"48 49" :the USER2 pattern is 48 49

>AUX:PATTERN:APPEND USER2,"AA BB 40" :append three bytes to USER2

>AUX:PATTERN:DATA? USER2 :report the new USER2 pattern

"48 49 AA BB 40" :the new bytes have been appended to USER2

>AUX:PATTERN:REPLACE USER2,3,"CC DD" :replace current USER2 string with CC DD starting at position 3

PATTERN (Continued)

User Message Selection **command/query**

>AUX:PATTERN:DATA? USER2 :report the new USER2 message

"48 49 CC DD 40"

>AUX:PATTERN:INSERT USER2,3,"00 01" :insert 00 and 01 in USER2 starting at position 3

>AUX:PATTERN:DATA? USER2 :report the new USER2 pattern

"48 49 00 01 CC DD 40"

>AUX:PATTERN:DELETE USER2,3,5 :delete five bytes from the USER2 beginning at position 3

>AUX:PATTERN:DATA? USER2 :report the new USER2 pattern

"48 49"

>AUX:PATTERN:SIZE? USER2 :report the current size of the USER2 pattern

2 :the USER2 message size is 2 bytes

PRGM_PATTERN

Programmable Data Pattern **command/query**

AUX:PRGM_PATTERN (bit pattern) :set a 3- to 24-bit programmable pattern

AUX:PRGM_PATTERN? :report current bit pattern

The **AUX:PRGM_PATTERN** command allows the user to program a data pattern. This data pattern is used when the DATA switch is in the PRGM position. The (bit pattern) can be any sequence of 3 to 24 ones (1's) and/or zeros (0's), corresponding to the desired binary bit pattern, with the left-most bit transmitted first.

The **AUX:PRGM_PATTERN?** query reports the user-programmed data pattern that is currently in use.

This remote control command is identical in function to **Auxiliary Function No. 33** when the FIREBERD is controlled from the front panel.

Example:

>AUX:PRGM_PATTERN 11010011 :set the programmable pattern to11010011

>AUX:PRGM_PATTERN? :print out the programmable pattern 11010011

RCV_CLK_POL

Receiver Clock Polarity **command/query**

AUX:RCV_CLK_POL [AUTO|NORMAL|INVERT] :set receiver clock polarity

AUX:RCV_CLK_POL? :report the current status of receiver clock polarity

The **AUX:RCV_CLK_POL** command sets the FIREBERD's receiver clock polarity. AUTO automatically detects, corrects, and maintains receiver clock polarity while error analysis is in progress; NORMAL sets receiver clock polarity to the current interface standard; INV sets receiver clock polarity to the inverse of the interface standard. AUTO is the default.

The **AUX:RCV_CLK_POL?** query reports the current generator clock polarity setting.

This remote control command is identical in function to **Auxiliary Function No. 2** when the FIREBERD is controlled from the front panel.

Example:

```
>AUX:RCV_CLK_POL AUTO   :set automatic receiver clock polarity
>AUX:RCV_CLK_POL?       :then display its status
AUTO
```

RCV_CLK_SEL

Select a Receiver Clock Source **command/query**

AUX:RCV_CLK_SEL [AUTO|ST|TT] :select a receiver clock source

AUX:RCV_CLK_SEL? :report the currently selected clock source

The **AUX:RCV_CLK_SEL** command selects the FIREBERD's receiver clock source. ST selects Send Timing (a DCE output); TT selects Terminal Timing (a DCE input); AUTO automatically selects ST if TT is not present in the current interface. AUTO is the default selection.

The **AUX:RCV_CLK_SEL?** query reports the currently selected clock source.

This remote control command is identical in function to **Auxiliary Function No. 7** when the FIREBERD is controlled from the front panel.

Example:

>AUX:RCV_CLK_SEL AUTO :set automatic selection of the receiver clock

>AUX:RCV_CLK_SEL? :display the receiver clock selection status

AUTO

RS232

RS-232 Printer/Remote Control Configuration command/query

AUX:RS232:BAUD [110|300|600|1200|2400|4800|9600] :set baud rate

AUX:RS232:DATA_BITS [7|8] :set data bits

AUX:RS232:PARITY [ODD|EVEN|NONE] :set parity bit

AUX:RS232? :report the current baud, data, and parity settings

AUX:RS232:BAUD? :report the current baud rate

AUX:RS232:DATA_BITS? :report the current data bits

AUX:RS232:PARITY? :report the current parity

This series of commands and queries are dedicated to RS-232 printer/remote control configuration.

The **AUX:RS232:BAUD** command is used to specify baud rates of 110, 300, 600, 1200, 2400, 4800 or 9600.

The **AUX:RS232:DATA_BITS** command is used to select 7 or 8 data bits.

The **AUX:RS232:PARITY** command is used to select odd, even, or no parity.

The **AUX:RS232?** query reports the current baud, data, and parity settings. The **AUX:RS232:BAUD?**, **AUX:RS232:DATA_BITS?**, and **AUX:RS232:PARITY?** queries individually report the current baud rate, data bits, and parity, respectively.

This remote control command is identical in function to **Auxiliary Function No. 38** when the FIREBERD is controlled from the front panel.

NOTE: These commands cannot change these parameters when RS-232 is used via remote control.

RS232 (Continued)

RS-232 Printer/Remote Control Configuration command/query

Example:

>AUX:RS232:BAUD?	:report the current baud rate
9600	:the current baud rate is 9600
>AUX:RS232:BAUD 300	:change the baud rate to 300
>AUX:RS232:PARITY?	:report the current parity setting
EVEN	:parity is even
>AUX:RS232:PARITY ODD	:change the parity to odd
>AUX:RS232?	:report the current baud, data, and parity settings
300, 7, ODD	

SINGLE_XMIT

Transmit a Single Pattern or Message **command/query**

AUX:SINGLE_XMIT [OFF|ON] :enable single or continuous transmission

AUX:SINGLE_XMIT? :report the current status

The **AUX:SINGLE_XMIT** command enables single transmit mode. OFF allows data to be transmitted continuously; ON allows only one pattern or message to be transmitted during a single test. OFF is the default selection.

The **AUX:SINGLE_XMIT?** query reports the current status of this activity.

This remote control command is identical in function to **Auxiliary Function No. 4** when the FIREBERD 6000 is controlled from the front panel.

Example:

```
>AUX:SINGLE_XMIT ON          :enable single transmit mode
>AUX:SINGLE_XMIT?            :display the current status of this
                              activity
ON
```

SITE_ID

Site Identification **command/query**

AUX:SITE_ID "(site identifier)" :set a site identifier

AUX:SITE_ID? :report the current site identifier

The **AUX:SITE_ID** command updates the site ID that appears with most status and results prints. The (site identifier) may be from 1 to 10 alphanumeric characters in length, and is always placed between quotation marks.

The **AUX:SITE_ID?** query reports the site identifier that is currently in effect.

This remote control command is identical in function to **Auxiliary Function No. 34** when the FIREBERD is controlled from the front panel.

Example:

>**AUX:SITE_ID?**	:report the current site ID
"Alpha"	:alpha is the current site ID
>**AUX:SITE_ID "BETA2"**	:change the current site ID to BETA2
>**AUX:SITE_ID?**	:report the current site ID
"BETA2"	

SYNCLOSS_ACT

Receiver Action on Synchronization Loss **command/query**

AUX:SYNCLOSS_ACT [CLEAR|HALT|CONT] :set the action on synchronization loss

AUX:SYNCLOSS_ACT? :report the current action on synchronization loss

The **AUX:SYNCLOSS_ACT** command sets the receiver action during the loss of synchronization. CLEAR resets all results counters and restarts the test when synchronization is reacquired. HALT freezes all test results for the duration of the synchronization loss; the test continues when synchronization is reacquired. CONT allows test results to accumulate while the receiver attempts to reacquire synchronization; this selection is only valid in SYNC and RECOVD timing modes. CLEAR is the default selection.

The **AUX:SYNCLOSS_ACT?** query reports the current action on synchronization loss.

This remote control command is identical in function to **Auxiliary Function No. 3** when the FIREBERD is controlled from the front panel.

Example:

>AUX:SYNCLOSS_ACT CLEAR :restart the test after reacquiring receiver synchronization

>AUX:SYNCLOSS_ACT? :display the current action on receiver synchronization loss

CLEAR

SYNCLOSS_THR

Synchronization Loss Error Threshold **command/query**

AUX:SYNCLOSS THR [LOW|MED|HIGH] :set the synchronization loss error rate threshold

AUX:SYNCLOSS THR? :report the current synchronization loss error rate threshold

The **AUX:SYNCLOSS_THR** command sets the error rate threshold required for the FIREBERD to declare a loss of synchronization. LOW sets the threshold to 100 errors in 1000 bits; MED sets the threshold to 250 errors in 1000 bits; HIGH sets the threshold to 20,000 errors in 100,000 bits.

The **AUX:SYNCLOSS_THR?** query reports the current synchronization loss threshold error rate.

This remote control command is identical in function to **Auxiliary Function No. 10** when the FIREBERD is controlled from the front panel.

Example:

```
>AUX:SYNCLOSS_THR MED      :set medium sync loss threshold
>AUX:SYNCLOSS_THR?         :report current sync loss threshold
MED
```

TIME

Set and Display Time — command/query

AUX:TIME "(time)" :set the time of day

AUX:TIME? :report the time of day

The **AUX:TIME** command sets the current time of day. The time is entered in 24-hour format, and is always placed between quotation marks; seconds are always set to zero.

The **AUX:TIME?** query reports the current time of day.

This remote control command is identical in function to **Auxiliary Function No. 60** when the FIREBERD is controlled from the front panel.

Example:

```
>AUX:TIME "13:45"          :set the time of day to 1:45 P.M.
>AUX:TIME?                 :report the current time of day
"13:45:00"
```

USR_SYNC_THR

User Message Sync Acquisition Threshold **command/query**

AUX:USR_SYNC_THR [PATTERN|BYTES] :set the synchronization acquisition threshold

AUX:USR_SYNC_THR? :report the byte length of the synchronization acquisition threshold

AUX:USR_SYNC_THR:BYTES (number) :set the byte length of the synchronization acquisition threshold

AUX:USR_SYNC_THR:BYTES? :report the byte length of the synchronization acquisition threshold

The **AUX:USR_SYNC_THR** command sets the synchronization acquisition threshold. PATTERN sets the length of the selected user message; BYTES sets the length by a specific number of bytes.

The **AUX:USR_SYNC_THR?** query reports the state of the synchronization acquisition threshold.

The **AUX:USR_SYNC_THR:BYTES** command sets the length of the synchronization acquisition threshold if it is set to BYTES. The valid range is 10-2048 bytes.

The **AUX:USR_SYNC_THR:BYTES?** query reports the byte length of the synchronous acquisition threshold if it is set to BYTES.

This remote control command is identical in function to **Auxiliary Function No. 45** when the FIREBERD is controlled from the front panel.

USR_SYNC_THR (Continued)

User Message Sync Acquisition Threshold **command/query**

Example:

>AUX:USR_SYNC_THR BYTES	:set the synchronization acquisition threshold to the length specified in the command immediately following
>AUX:USR_SYNC_THR:BYTES 256	:set the synchronization threshold to 256 bytes
>AUX:USR_SYNC_THR:BYTES?	:display the length in bytes of the synchronization acquisition threshold

256

CONFIGURATION COMMANDS

The FIREBERD 6000 features a set of commands and queries that can remotely configure the instrument's switches and LEDs. These commands and queries are:

Commands	Queries
CONFIG:ANA_MODE	CONFIG:ANA_MODE?
	CONFIG:CHAR_FORMAT?
CONFIG:CHAR_FORMAT:DATA_BITS	CONFIG:CHAR_FORMAT:DATA BITS?
CONFIG:CHAR_FORMAT:PARITY	CONFIG:CHARFORMAT:PARITY?
CONFIG:CHAR_FORMAT:STOP_BITS	CONFIG:CHAR_FORMAT:STOP_BITS?
CONFIG:CHAR_ERR_INSERT	CONFIG:CHARFORMAT:ERR_INSERT?
CONFIG:DM	CONFIG:DM?
CONFIG:DSR	CONFIG:DSR?
CONFIG:DTR	CONFIG:DTR?
CONFIG:FREQ	CONFIG:FREQ?
CONFIG:GEN_CLK	CONFIG:GEN_CLK?
CONFIG:LOOP	CONFIG:LOOP?
CONFIG:PATTERN	CONFIG:PATTERN?
CONFIG:RLSD	CONFIG:RLSD?
CONFIG:RR	CONFIG:RR?
CONFIG:RTS	CONFIG:RTS?
CONFIG:SELF_LOOP	CONFIG:SELF_LOOP?
CONFIG:TEST_INT	CONFIG:TEST_INT?
CONFIG:TIMING_MODE	CONFIG:TIMING_MODE?
CONFIG:TPZ	CONFIG:TPZ?

ANA_MODE

Analysis Mode **command/query**

CONFIG:ANA_MODE [SINGLE|CONTINUOUS] :set the FIREBERD analysis mode

CONFIG:ANA_MODE? :report the current analysis mode

The **CONFIG:ANA_MODE** command sets the FIREBERD 6000 to either single or continuous analysis mode. In SINGLE analysis mode, the actual length of the test is determined by the CONFIG:TEST_INT remote control command; in CONTINUOUS analysis mode, a test runs indefinitely and continues to accumulate test results.

The **CONFIG:ANA_MODE?** query reports the current mode of analysis.

This remote control command is identical in function to the ANALYSIS MODE switch, located on the FIREBERD 6000 front panel.

Example:

>CONFIG:ANA_MODE? :report the current mode of analysis

SINGLE :the current analysis mode is SINGLE

>CONFIG:ANA_MODE CONTINUOUS
:change the analysis mode to CONTINUOUS

CHAR_FORMAT

Asynchronous Character Format **command/query**

CONFIG:CHAR_FORMAT:
DATA_BITS [5|6|7|8]* :set the number of data bits

CONFIG:CHAR_FORMAT:
PARITY|ODD |EVEN|NONE] :set the parity

CONFIG:CHAR_FORMAT:
STOP_BITS [1|1.5|2] :set the number of stop bits

CONFIG:CHAR_FORMAT:
DATA_BITS? :report the current number of data bits

CONFIG:CHAR_FORMAT:
PARITY? :report the current parity setting

CONFIG:CHAR_FORMAT:
STOP_BITS? :report the current number of stop bits

CONFIG:CHAR_FORMAT? :report the current data bit, parity, and stop bit settings

This series of commands and queries is dedicated to selection of the asynchronous character format.

The **CONFIG:CHAR_FORMAT** commands set data bits, parity, and stop bits.

The **CONFIG:CHAR_FORMAT?** query reports the current data bit, parity, and stop bit settings. Note that current settings can also be queried individually.

This remote control command is identical in function to the CHAR FORMAT category of the FIREBERD 6000's MENU switch.

*5=BCD
6=BAUDOT
7=ASCII
8=EBCDIC

CHAR_FORMAT (Continued)

Asynchronous Character Format **command/query**

Example:

```
>CONFIG:CHAR_FORMAT?                    :report the data bit, parity,
                                         and stop bit settings
7, ODD, 1.5
>CONFIG:CHAR_FORMAT:PARITY EVEN         :change the parity setting
                                         to even
>CONFIG:CHAR_FORMAT:PARITY?             :report the new parity setting
EVEN
```

DM

DM Line Control — command/query

CONFIG:DM [ON|OFF] :set the DM signaling lead

CONFIG:DM? :report the current state of the DM signaling lead

The **CONFIG:DM** command sets the state of the RS-449 DM signaling lead (i.e., ON or OFF). It is only valid when the FIREBERD 6000 is operating in DCE mode.

The **CONFIG:DM?** query reports the current state of the DM signaling lead.

Example:

>CONFIG:DM? :report the current state of the DM signaling lead

OFF

>CONFIG:DM ON :turn the DM signaling lead

DSR

DSR Line Control — command/query

CONFIG:DSR [ON|OFF] :set the DSR signaling lead

CONFIG:DSR? :report the current state of the DSR signaling lead

The **CONFIG:DSR** command sets the state of the RS-232 DSR signaling lead (i.e., ON or OFF). It is only valid when the FIREBERD 6000 is operating in DCE mode.

The **CONFIG:DSR?** query reports the current state of the DSR signaling lead.

Example:

>CONFIG:DSR? :report the current state of the DSR signaling lead

ON

>CONFIG:DSR OFF :turn the DSR signaling lead off

DTR

DTR Line Control — command/query

CONFIG:DTR [ON|OFF] :set the DTR signaling lead

CONFIG:DTR? :report the current state of the DTR signaling lead

The **CONFIG:DTR** command sets the state of the DTR (for RS-232) or the TR (for RS-449) signaling lead (i.e., ON or OFF). It is only valid when the FIREBERD 6000 is operating in DTE mode.

The **CONFIG:DTR?** query reports the current state of the DTR signaling lead.

Example:

>CONFIG:DTR? :report the current state of the DTR signaling lead

OFF

>CONFIG:DTR ON :turn the DTR signaling lead on

ERR_INSERT

Logic Error Insertion **command/query**

CONFIG:ERR_INSERT
[OFF|SINGLE|1.0E-3] :set error insertion status

CONFIG:ERR_INSERT? :report the current error insertion status

The **CONFIG:ERR_INSERT** command sets the desired error insertion status. OFF turns automatic error insertion off; SINGLE causes a single error to be inserted into the generated data; 1.0E-3 causes one error to be inserted per 1000 bits of generated data.

NOTE: Single error insertion terminates 1.0E3 insertion, if it is in effect.

The **CONFIG:ERR_INSERT?** query reports the current error insertion status.

This remote control command is identical in function to the ERROR INSERT switch on the FIREBERD 6000's front panel.

Example:

>CONFIG:ERR_INSERT OFF :turn off automatic error insertion
>CONFIG:ERR_INSERT? :report the current error insertion status
OFF
>CONFIG:ERR_INSERT 1.0E-3 :enable automatic error insertion
>CONFIG:ERR_INSERT SINGLE :insert a single error
>CONFIG:ERR_INSERT? :report the current error insertion status
OFF :the 1.0E-3 rate is terminated

FREQ

Set Synthesizer Frequency **command/query**

CONFIG:FREQ (frequency) :set the synthesizer frequency

CONFIG:FREQ? :report the current synthesizer frequency

The **CONFIG:FREQ** command sets the FIREBERD 6000's internal frequency synthesizer in kHz. The (frequency) can be any valid kHz value within the range .05 kHz and 15000 kHz.

The **CONFIG:FREQ?** query reports the current synthesizer frequency.

This remote control command is identical in function to the SYNTH FREQ category of the FIREBERD 6000's MENU switch.

Certain combinations of synthesizer, data pattern, and timing mode settings are not valid. If the FIREBERD's new frequency setting is invalid, a contention error appears in the error buffer and the previous frequency setting is retained.

To reduce the possibility of an invalid configuration, the FIREBERD delays processing of this command until all other commands of the program message are read. The delayed commands are considered at the same time; an error is flagged only if this final configuration is invalid.

See Section 6.5 of the *FIREBERD 6000 Operating Manual* for a description of the settings which are invalid.

Example:

>CONFIG:FREQ? :report the current synthesizer frequency

9.6
>CONFIG:FREQ 1544 :set the frequency synthesizer to 1544 kHz

GEN_CLK

Generator Clock Select **command/query**

CONFIG:GEN_CLK [SYNTH|INTF|BNC] :set the generator clock

CONFIG:GEN_CLK? :report the current generator clock setting

The **CONFIG:GEN_CLOCK** command sets the FIREBERD 6000's generator clock. SYNTH sets the FIREBERD's built-in frequency synthesizer as the source of signal timing; INTF selects the clock originating from the network or the clock in the interface module as the source of signal timing; BNC selects an external timing source. If BNC is selected, the external timing source must be connected to the GEN CLK IN BNC connector on the FIREBERD 6000's rear panel.

The **CONFIG:GEN_CLOCK?** query reports the current setting of the generator clock.

This remote control command is identical in function to the GEN CLK category on the FIREBERD 6000's MENU switch.

Example:

>CONFIG:GEN_CLK SYNTH :select the synthesizer as the source of signal timing

>CONFIG:FREQ? :report the current synthesizer frequency

9.6

>CONFIG:GEN_CLK? :report the current generator clock setting

SYNTH

LOOP

Generator Loop Codes — command

CONFIG:LOOP [UP|DOWN] :generate loop up or loop down codes

When using a T1 interface, the **CONFIG:LOOP** command generates LOOP UP or LOOP DOWN code. In such instances, loop code transmission is valid only when testing out-of-service.

When using the DDS DS0A/B Interface (Model 30678), the **CONFIG:LOOP** command also generates LOOP UP or LOOP DOWN code. To send either loop code, the interface must be placed in either alternating or latching loopback mode.

NOTE: The **CONFIG:LOOP DOWN** command should always be used at the conclusion of any loopback test to remove the effect of loop-up code (transmitted via the **CONFIG:LOOP UP** command) on the circuit under test.

Example:

>CONFIG:LOOP UP :generate loop up code
Transmitting Loop Code
> :loop-up code transmission is successful

.
.
.

>CONFIG:LOOP DOWN :loop down code is transmitted at the conclusion of the test to return the circuit to normal operation

Transmitting Loop Code
> :loop-down code transmission is successful

PATTERN

Select a Data Pattern — command/query

CONFIG:PATTERN [MARK|1:1|63|511|2047|2 15-1|2 20-1|2 23-1|QRSS|PRGM|FOX|USER] :select a data pattern

CONFIG:PATTERN? :return the current data pattern

The **CONFIG:PATTERN** command selects a data pattern to be generated by the FIREBERD. The data pattern selected may be any one of the following:

MARK	all Marks
1:1	alternating Mark and Space
63	63-bit pseudorandom pattern
511	511-bit pseudorandom pattern
2047	2047-bit pseudorandom pattern
$2^{15}-1$	$2^{15}-1$ pseudorandom pattern
$2^{20}-1$	$2^{20}-1$ pseudorandom pattern
$2^{23}-1$	$2^{23}-1$ pseudorandom pattern
QRSS	QRSS pattern
PRGM	programmable pattern
FOX	QUICK BROWN FOX message
USER	user-defined message

The **CONFIG:PATTERN?** query reports the currently selected data pattern.

This remote control command is identical in function to the DATA switch on the FIREBERD 6000's front panel.

Section 3 of the *FIREBERD 6000 Operating Manual* fully describes all data patterns generated by the FIREBERD 6000.

Certain combinations of synthesizer, data pattern, and timing mode settings are not valid. If the FIREBERD's new pattern setting is invalid, a contention error appears in the error buffer and the previous pattern setting is retained.

PATTERN (Continued)

Select a Data Pattern — command/query

To reduce the possibility of an invalid configuration, the FIREBERD delays processing of this command until all other commands of the program message are read. The delayed commands are considered at the same time; an error is flagged only if this final configuration is invalid.

See Section 6.5 of the *FIREBERD 6000 Operating Manual* for a description of the settings which are invalid.

Example:

```
>CONFIG:PATTERN?              :report the current data pattern
511
>CONFIG:PATTERN QRSS          :select the QRSS data pattern
```

RLSD

RLSD Line Control — command/query

CONFIG:RLSD [ON|OFF] :set the RLSD signaling lead

CONFIG:RLSD? :report the current state of the RLSD signaling lead

The **CONFIG:RLSD** command sets the state of the RS-232 RLSD signaling lead (i.e., ON or OFF). It is only valid when the FIREBERD 6000 is operating in DCE mode.

The **CONFIG:RLSD?** query reports the current state of the RLSD signaling lead.

Example:

>CONFIG:RLSD? :report the current state of the RLSD signaling lead

ON

>CONFIG:RLSD OFF :turn the RLSD signaling lead off

RR

RR Line Control **command/query**

CONFIG:RR [ON|OFF] :set the RR signaling lead

CONFIG:RR? :report the current state of the RR signaling lead

The **CONFIG:RR** command sets the state of the RS-449 RR signaling lead (i.e., ON or OFF). It is only valid when the FIREBERD 6000 is operating in DCE mode.

The **CONFIG:RR?** query reports the current state of the RR signaling lead.

Example:

>CONFIG:RR? :report the current state of the RR signaling lead

OFF

>CONFIG:RR ON :turn the RLSD signaling lead on

RTS

RTS Line Control — command/query

CONFIG:RTS [ON|OFF] :set the RTS signaling lead

CONFIG:RTS? :report the current state of the RTS signaling lead

The **CONFIG:RTS** command sets the state of the RS-232 RTS or the RS-449 signaling lead (i.e., ON or OFF). It is only valid when the FIREBERD 6000 is operating in DTE mode.

The **CONFIG:RTS?** query reports the current state of the RTS signaling lead.

Example:

>CONFIG:RTS? :report the current state of the RTS signaling lead

ON

>CONFIG:RTS OFF :turn the RTS signaling lead off

SELF_LOOP

Self-Loop Select **command/query**

CONFIG:SELF_LOOP [ON|OFF] :place the FIREBERD in self-loop mode

CONFIG:SELF_LOOP? :report the current self-loop state

The **CONFIG:SELF_LOOP** command takes the FIREBERD 6000 in and out of self-loop mode (i.e., ON and OFF).

The **CONFIG:SELF_LOOP?** query reports the current self-loop state.

This remote control command is identical in function to the SELF LOOP switch on the FIREBERD 6000's front panel.

Example:

>CONFIG:SELF_LOOP? :report the current self-loop state
OFF :self-loop is OFF
>CONFIG:SELF_LOOP ON :turn self-loop ON

TEST_INT

Select Test Interval **command/query**

CONFIG:TEST_INT ["1.0E4"|"1.0E5" "1.0E6"|"1.0E7"|"1.0E8"|"1.0E9" "1.0E10"|("hh:mm:ss")] :select the number of bits in the test interval or select a timed test interval

CONFIG:TEST_INT? :report the current test interval

The **CONFIG:TEST_INT** command selects the test interval when SINGLE analysis mode is selected using the **CONFIG:ANA_MODE** command. This interval may be a number of bits within the range 1.0E4-1.0E10, and must be placed within quotation marks. The test interval may also be a time period, expressed in hours, minutes, and seconds within quotation marks.

The **CONFIG:TEST_INT?** query reports the current test interval.

This remote control command is identical in function to the TEST INTERVAL category of the FIREBERD 6000's MENU switch.

Example:

>CONFIG:TEST_INT? :report the current test interval

"1.0E6" :the current test interval is 1.0E6 bits

>CONFIG:TEST_INT "00:10:00" :set a timed test interval of 10 minutes

>CONFIG:TEST_INT? :report the current test interval

"00:10:00"

TIMING_MODE

Select Timing Mode **command/query**

CONFIG:TIMING_MODE [SYNC|ASYNC|RECOVD] :select a timing mode

CONFIG:TIMING_MODE? :report the currently selected timing mode

The **CONFIG:TIMING_MODE** command selects a timing mode. SYNC selects synchronous timing; ASYNC selects asynchronous timing; RECOVD selects recovered timing. Each is discussed in detail in Section 3 of the *FIREBERD 6000 Operating Manual.*

The **CONFIG:TIMING_MODE?** query reports the currently selected timing mode. When the current interface does not require a timing mode setting, the message N/A is returned.

This remote control command is identical in function to the TIMING MODE switch on the FIREBERD 6000's front-panel.

Certain combinations of synthesizer, data pattern, and timing mode settings are not valid. If the FIREBERD's new timing mode setting is invalid, a contention error appears in the error buffer and the previous timing mode setting is retained.

To reduce the possibility of an invalid configuration, the FIREBERD delays processing of this command until all other commands of the program message are read. The delayed commands are considered at the same time; an error is flagged only if this final configuration is invalid.

See Section 6.5 of the *FIREBERD 6000 Operating Manual* for a description of the settings which are invalid.

TIMING_MODE (Continued)

Select Timing Mode **command/query**

Example:

>CONFIG:TIMING_MODE?	:report the current timing mode
SYNC	:synchronous timing is currently selected
>CONFIG:TIMING_MODE ASYNC	:change the current timing mode to asynchronous

TPZ

Set State of Rear-Panel Test Point **command/query**

CONFIG:TPZ [HIGH|LOW] :set the state of the rear-panel test point

CONFIG:TPZ? :report the state of the rear-panel test point

The **CONFIG:TPZ** command selects the state of the rear-panel test point. HIGH raises the test point to +5 volts; LOW lowers the test point to 0 volts.

The **CONFIG:TPZ?** query reports the current state of the rear-panel test point.

Example:

>CONFIG:TPZ? :report the current state of the rear-panel test point

LOW :the rear-panel test point is at 0 volts

>CONFIG:TPZ HIGH :change the state of the rear-panel test point

INTERFACE COMMANDS

Interface commands and queries remotely control the FIREBERD 6000's interfaces. The following interfaces are described in this section:

- Internal RS-232 Interface
- DDS DS0/DS0A Interface (Model 30481)
- DS1/T1/D4/ESF Interface (Model 40460)
- DS1/T1/D4/ESF SLC-96 Interface (Model 40540)
- CCITT G.703 8448 kb/s Interface (Model 30524)
- CCITT G.704 2048 kb/s Interface (Model 30609)
- CCITT G.703 kb/s Interface (Model 30608)
- ISDN S/T TE Interface (Model 41075)
- DDS Local Loop Interface (Model 41131)
- DS1/FT1 Interface (Model 41440)
- RS-449/530/MIL Interface (Model 41400)
- DDS DS0A/DS0B Interface (Model 30678)

In this section, the interfaces are described *in the order in which they appear in the list.*

Selecting the Internal RS-232 Interface For Remote Control

To select the internal RS-232 interface for remote control, issue the following command prior to executing the interface-specific commands and queries:

>INTF:SELECT INT232

Selecting A Modular Interface For Remote Control

To select a modular interface for remote control, issue the following command before executing any interface-specific commands and queries:

>INTF:SELECT EXT

Selecting A Modular Interface Installed In An ISU-6000

If a modular interface is installed in an ISU-6000, specify the ISU and slot numbers as follows:

>INTF:SELECT:ISU1,3

EXAMPLE: The desired modular interface is in ISU #1, Slot #3.

INT232

Internal RS-232 Interface Control **command/query**

INTF:INT232 [DTE|DCE]	:emulate DTE or DCE
INTF:INT232:DCE:RI [ON|OFF]	:control RI signal
INTF:INT232:DCE:CTS [ON|OFF]	:control CTS signal
INTF:INT232?	:report current emulation
INTF:INT232:DCE:RI?	:report current RI signal status
INTF:INT232:DCE:CTS?	:report current CTS signal status

This series of commands and queries enables remote control of the FIREBERD 6000's built-in RS-232 Interface.

For more information on the RS-232 interface, please refer to Section 8 of the *FIREBERD 6000 Operating Manual.*

Example:

>INTF:INT232?	:report the current emulation
DTE	
>INTF:INT232 DCE	:enable DCE emulation
>INTF:INT232:DCE:RI?	:report the current RI signal status
OFF	
>INTF:INT232:DCE:RI ON	:turn the RI signal on

DS0

DDS DS0A Interface Control **command/query**

INTF:DS0:MODE [DS0|DS0A] :select the interface mode

INTF:DS0:DATA_TYPE [BIPOLAR|LOGIC_NEAR|LOGIC_FAR] :select a data format

INTF:DS0:LOOP_TYPE [DSU|CHAN|OCU|NONE] :select loopback

INTF:DS0:RATE [2.4|4.8|9.6|56] :select a data rate

INTF:DS0:CODE_ONLY :select loop code

INTF:DS0:MODE? :report the current interface mode

INTF:DS0:LOOP_TYPE? :report the current loop type

INTF:DS0:RATE? :report the current rate

This series of commands and queries enables remote control of the DDS DS0/DS0A Interface.

For specifics on interface settings, please refer to the *DDS DS0/DS0A Interface Manual*.

Example:

>INTF:DS0:RATE? :report the current data rate DS0A
>INTF:DSO:RATE 9.6 :set the data rate to 9.6

DS1_FE

DS1/T1/D4/ESF Control	command/query
INTF:DS1_FE:RESULT_MODE [STD|LIV]	:select result mode
INTF:DS1_FE:THRU_MODE [ON|OFF]	:select thru-data mode
INTF:DS1_FE:INPUT [TERM|BRIDGE|DSX_MON]	:select input impedance
INTF:DS1_FE:CODING [AMI|B8ZS]	:select line code
INTF:DS1_FE:FRAMING [OFF|D4|FE]	:select framing format
INTF:DS1_FE:LBO [0|-7.5|-15]	:select line buildout (dB)
INTF:DS1_FE:ERR_INSERT [OFF|SIN|RATE (rate)]	:select BPV insertion
INTF:DS1_FE:ERR_RATE (rate string)	:select the user-specified BPV insertion rate (xE-y, where x = 1 to 9 and y = 3 to 9)
INTF:DS1_FE:RESULT_MODE?	:report current result mode
INTF:DS1_FE:THRU_MODE?	:report current thru-data mode
INTF:DS1_FE:INPUT?	:report current input impedance
INTF:DS1_FE:CODING?	:report current line code
INTF:DS1_FE:FRAMING?	:report current framing format
INTF:DS1_FE:LBO?	:report current line buildout (dB)

DS1_FE

DS1/T1/D4/ESF Interface Control — command/query

INTF:DS1_FE:ERR_INSERT? :report current BPV insertion status

INTF:DS1_FE:ERR_RATE? :report current BPV insertion rate

This series of commands and queries enables remote control of the DS1/T1/D4/ESF Interface.

For specifics on interface settings, please refer to the *DS1/T1/D4/ESF Interface Manual*.

Example:

>INTF:DS1 FE:CODING? :report the code setting
AMI
>INTF:DS1_FE:CODING B8ZS :change the code setting to B8ZS

DS1_T1

DS1/T1/D4/ESF/SLC-96 Interface Control — command/query

INTF:DS1_T1: RESULT_MODE [STD|LIV]	:select result mode
INTF:DS1_T1:THRU_MODE [OFF|TLB|LLB]	:select thru-data mode
INTF:DS1_T1:INPUT [TERM| BRIDGE|DSX_MON]	:select input impedance
INTF:DS1_T1: CODING [AMI|B8ZS]	:select line code
INTF:DS1_T1: FRAMING [OFF|D4|FE|SLC]	:select framing format
INTF:DS1_T1: LBO [0|-7.5|-15]	:select line buildout (dB)
INTF:DS1_T1:ERR_INSERT [OFF|BPV|LOGIC|L+BPV| SINGLE|RATE (rate)]	:select error insertion
INTF:DS1_T1:ERR_RATE (rate string)	:select the user-specified error insertion rate (xE-y, where x = 1 to 9 and y = 3 to 9)
INTF:DS1_T1:RESULT_MODE?	:report current result mode
INTF:DS1_T1:THRU_MODE?	:report current thru data mode
INTF:DS1_T1:INPUT?	:report current input impedance
INTF:DS1_T1:CODING?	:report current line code
INTF:DS1_T1:FRAMING?	:report current framing format

DS1_T1

DS1/T1/D4/ESF/SLC-96 Interface Control command/query

INTF:DS1_T1:LBO? :report current line build out (dB)

INTF:DS1_T1:ERR_INSERT? :report current error insertion status

INTF:DS1_T1:ERR_RATE? :report current error insertion rate

This series of commands and queries enables remote control of the DS1/T1/D4/ESF/SLC-96 Interface.

For specifics on interface settings, please refer to the *DS1/T1/D4/ESF/SLC-96 Interface Manual*.

Example:

>INTF:DS1_T1:ERR_INSERT? :report the error insertion status
L+BPV
>INTF:DS1_T1:ERR_INSERT OFF :turn error insertion off

8MG703

G.703 8448 kb/s Interface Control — command/query

INTF:8MG703:TIMING [XTAL|RECOV] :select a transmit timing reference

INTF:8MG703:RESULT_MODE [NORMAL|BPV] :select an error detection mode

INTF:8MG703:INPUT [TERM|BRIDGE] :select an input impedance

INTF:8MG703:TIMING? :report the current transmit timing reference

INTF:8MG703:RESULT_MODE? :report the current error detection mode

INTF:8MG703:INPUT? :report the current input impedance

This series of commands and queries enables remote control of the CCITT G.703 8448 kb/s Interface.

For specifics on interface settings, please refer to the *CCITT G.703 8448 kb/s Interface Manual*.

Example:

>INTF:8MG703:RESULT_MODE? :report the current result mode

BPV

>INTF:8MG703:RESULT_MODE NORMAL :change the result mode to normal

2MG704

G.704 2048 kb/s Interface Control — command/query

INTF:2MG704:FRAMING [OFF|ON] :enable or disable framing

INTF:2MG704:FRAMING: CRC4 [OFF|ON] :enable or disable CRC-4 multiframe

INTF:2MG704:FRAMING: TS16 [OFF|ON] :enable or disable TS-16 multiframe

INTF:2MG704:RESULT_MODE [STD|LIVE] :select results mode

INTF:2MG704:THRU_MODE [OFF|ON] :enable or disable thru-loopback mode

INTF:2MG704:ERR_INSERT [OFF|SINGLE\RATE] :select error insertion

INTF:2MG704:ERR_RATE (rate string) :specify an error insertion rate (xE-y, where x = 1 to 9 and y = 3 to 9)

INTF:2MG704:INPUT [TERM|BRIDGE] :select the 75-ohm or 120-ohm interface input impedance

INTF:2MG704:CODING [AMI|HDB3] :select code standard

INTF:2MG704:TIMING [XTAL|RECOV] :select timing source

INTF:2MG704:FRAMING? :report current framing status

2MG704 (Continued)

G.704 2048 kb/s Interface Control	command/query
INTF:2MG704:FRAMING:CRC4?	:report current CRC-4 multiframe status
INTF:2MG704:FRAMING:TS16?	:report current TS-16 multiframe status
INTF:2MG704:RESULT_MODE?	:report the current result mode
INTF:2MG704:THRU_MODE?	:report the current thru-mode status
INTF:2MG704:ERR_INSERT?	:report the current error insertion status
INTF:2MG704:ERR_RATE?	:report the current error rate
INTF:2MG704:INPUT?	:report the current input impedance
INTF:2MG704:CODING?	:report the current code standard
INTF:2MG704:TIMING?	:report the current timing source

This series of commands and queries enables remote control of the CCITT G.704 2048 kb/s Interface.

For specifics on interface settings, please refer to the *CCITT G.704 2048 kb/s Interface Manual.*

Example:

>INTF:2MG704:FRAMING:CRC4? :report the current CRC-4 multiframe status

OFF

>INTF:2MG704:FRAMING:CRC4 ON

:turn CRC-4 multiframing on

2MG704 (Continued)

G.704 2048 kb/s Interface Control	command/query
INTF:2MG704:FRAMING [OFF|ON]	:enable or disable framing
INTF:2MG704:FRAMING:CRC4 [OFF|ON]	:enable or disable CRC-4 multiframe
INTF:2MG704:FRAMING:TS16 [OFF|ON]	:enable or disable TS-16 multiframe
INTF:2MG704:RESULT_MODE [STD|LIVE]	:select results mode
INTF:2MG704:THRU_MODE [OFF|ON]	:enable or disable thru-loopback mode
INTF:2MG704:ERR_INSERT [OFF|SINGLE|RATE]	:select error insertion
INTF:2MG704:ERR_RATE (rate string)	:specify an error insertion rate (xE-y, where x = 1 to 9 and y = 3 to 9)
INTF:2MG704:INPUT [TERM|BRIDGE]	:select the 75-ohm or 120-ohm interface input impedance
INTF:2MG704:CODING [AMI|HDB3]	:select code standard
INTF:2MG704:TIMING [XTAL|RECOV]	:select timing source
INTF:2MG704:FRAMING?	:report current framing status

2MG704 (Continued)

G.704 2048 kb/s Interface Control — command/query

INTF:2MG704:FRAMING:CRC4?	:report current CRC-4 multiframe status
INTF:2MG704:FRAMING:TS16?	:report current TS-16 multiframe status
INTF:2MG704:RESULT_MODE?	:report current result mode
INTF:2MG704:THRU_MODE?	:report current thru-mode status
INTF:2MG704:ERR_INSERT?	:report current error insertion status
INTF:2MG704:ERR_RATE?	:report current error rate
INTF:2MG704:INPUT?	:report current input impedance
INTF:2MG704:CODING?	:report current code standard
INTF:2MG704:TIMING?	:report current timing source

This series of commands and queries enables remote control of the CCITT G.704 2048 kb/s Interface.

For specifics on interface settings, please refer to the *CCITT G.704 2048 kb/s Interface Manual.*

Example:

>INTF:2MG704:FRAMING:CRC4?	:report the current CRC-4 multiframe status
>INTF:2MG704:FRAMING:CRC4 ON	:turn CRC-4 multiframing on

64KG703

G.703 64 kb/s Interface Control	command/query
INTF:64KG703:MODE [CODIR|CENTRA|CONTRA]	:select interface mode
INTF:64KG703:MODE: CONTRA [DTE|DCE]	:select contra DTE or DTE interface mode
INTF:64KG703:OCTET_TIMING [OFF|ON]	:enable or disable octet timing
INTF:64KG703:INPUT [TERM|BRIDGE]	:select input impedance
INTF:64KG703:THRU_MODE [OFF|ON]	:enable or disable thru-loopback mode
INTF:2MG704:ERR_INSERT [OFF|SINGLE|RATE]	:select error insertion
INTF:2MG704:ERR_RATE (rate string)	:specify an error insertion rate (xE-y, where x = 1 to 9 and y = 3 to 9)
INTF:64KG703:MODE?	:report current interface mode
INTF:64KG703:MODE:CONTRA?	:report current contra mode (DTE or DCE)
INTF:64KG703:OCTET_TIMING?	:report current octet timing status
INTF:64KG703:INPUT?	:report current input impedance
INTF:64KG703:THRU_MODE?	:report current thru-loopback status

64KG703 (Continued)

G.703 64 kb/s Interface Control — command/query

INTF:64KG703:ERR_INSERT? :report error insertion rate

INTF:64KG703:ERR_RATE? :report the current error rate

This series of commands and queries enables remote control of the CCITT G.703 64 kb/s Interface.

For specifics on interface settings, please refer to the *CCITT G.703 64 kb/s Interface Manual.*

Example:

>INTF:64KG703:INPUT? :report the current line termination

TERM
>INTF:64KG703:INPUT:BRIDGE :set line termination to BRIDGE

ISDN

ISDN S/T TE Interface	command/query
INTF:ISDN_VOICE:CALL: MEMORY	:place a voice call to the number stored in memory
INTF:ISDN_VOICE:CALL: LAST	:place a voice call on the last dialed number
INTF:ISDN_VOICE:CALL: NUMBER (phone number)	:place a voice call to the phone number specified
INTF:ISDN_S:VOICE: DISCONNECT	:disconnect a voice call
INTF:ISDN_S:VOICE: ANSWER	:answer an incoming voice call
INTF:ISDN_S:VOICE: TX_CHANNEL [ANY|B1|B2]	:select a channel for voice calls
INTF:ISDN_S:VOICE:LAW [u-law|a-law]	:select voice channel encoding
INTF:ISDN_S:DATA:CALL: MEMORY	:place a data call to the number stored in memory
INTF:ISDN_S:DATA:CALL: LAST	:place a data call on the last dialed number
INTF:ISDN_S:DATA:CALL: NUMBER (phone number)	:place a data call to the phone number specified

ISDN (Continued)

ISDN S/T TE Interface	command/query
INTF:ISDN_S:DATA: DISCONNECT	:disconnect a data call
INTF:ISDN_S:DATA: ANSWER [AUTO|IGNORE]	:answer or ignore an incoming data call
INTF:ISDN_S:VOICE: TX_RATE[64kbu|64kbr|56kb]	:select the data transmission rate
INTF:ISDN_S:DATA:TX_CHANNEL [ANY|B1|B2]	:select a channel for data calls
INTF:ISDN_S:MEMORY (number) (phone number)	:place a phone number in memory
INTF:ISDN_S:SIGNAL [ACTIVE|INACTIVE]	:set signal status
INTF:ISDN_S:TERM [TERMINATE|UNTERMINATE]	:set termination status
INTF:ISDN_VOICE:CALL:LAST?	:report the last voice call number dialed
INTF:ISDN_S:VOICE:TX_CHANNEL?	:report current transmit voice channel
INTF:ISDN_S:VOICE:LAW?	:report current voice channel encoding
INTF:ISDN_S:DATA:CALL:LAST?	:report the last data call number dialed

ISDN (Continued)

ISDN S/T TE Interface	command/query
INTF:ISDN_S:DATA:ANSWER?	:report whether incoming data calls are answered or ignored
INTF:ISDN_S:VOICE:TX_RATE?	:report current data transmission rate
INTF:ISDN_S:DATA:TX_CHANNEL?	:report current transmit channel for data calls
INTF:ISDN_S:MEMORY?	:report current phone number in memory
INTF:ISDN_S:SIGNAL?	:report current signal status
INTF:ISDN_S:TERM?	:report current termination status
INTF:ISDN_S:REVISION?	:report the revision of the ISDN interface

This series of commands and queries enables remote control of the ISDN S/T TE Interface.

For specifics on interface settings, please refer to the *ISDN S/T TE Interface Manual*.

Example:

>INTF:ISDN_S:VOICE:TX_CHANNEL?

:report the current transmission channel for voice calls

ANY
>INTF:ISDN_S:VOICE:TX_CHANNEL B1

:transmit voice calls on B1 only

DDSLL

DDS Local Loop Interface Control **command/query**

INTF:DDSLL:PRIMARY_RATE [2.4|4.8|9.6|19.2|56.0] :select primary channel data rate

INTF:DDSLL:TX_LEVEL [0|-3|-6|-9] :select transmit level in dB

INTF:DDSLL:TX_CLOCK [RXTAL|RECOVERY] :select transmit clock source

INTF:DDSLL:ERR_INSERT [OFF|BPV|LOGIC|L+BPV|SINGLE|RATE] :select error insertion operation

INTF:DDSLL:ERR_RATE (rate string) :specify an error insertion rate (xE-y, where x = 1 to 9 and y = 3 to 9)

INTF:DDSLL:SECONDARY_CHANNEL [OFF|ON] :select secondary channel operation

INTF:DDSLL:ANALYSIS_CHANNEL [PRIMARY|SECONDARY] :select a channel for analysis

INTF:DDSLL:ALTERNATE_PATTERN [511|2047|IDLE] :transmit an alternate data pattern

INTF:DDSLL:PRIMARY_RATE? :report current primary data rate

INTF:DDSLL:TX_LEVEL? :report current transmit level

INTF:DDSLL:TX_CLOCK? :report transmit clock source

DDSLL (Continued)

DDS Local Loop Interface Control	command/query

INTF:DDSLL:ERR_INSERT? :report current error insertion operation

INTF:DDSLL:ERR_RATE? :report current error insertion rate

INTF:DDSLL:SECONDARY_CHANNEL? :report secondary channel status

INTF:DDSLL:ANALYSIS_CHANNEL? :report current channel being analyzed

INTF:DDSLL:ALTERNATE_ PATTERN? :report current alternate pattern selection

This series of commands and queries enables remote control of the DDS Local Loop Interface.

For specifics on interface settings, please refer to the *DDS Local Loop Interface Manual.*

Example:

>INTF:DDSLL:ANALYSIS_CHANNEL? :report the channel being analyzed

PRIMARY

>INTF:DDSLL:ANALYSIS_CHANNEL SECONDARY

:set secondary channel for analysis

T1_FT1

T1/FT1 Interface Control — command/query

INTF:T1_FT1:CONFIG:CODE [AMI|B8ZS] :select coding

INTF:T1_FT1:CONFIG:FRAME [OFF|D4|ESF|SLC] :select framing

INTF:T1_FT1:CONFIG:INPUT [TERM|BRIDGE|DSX_MON] :select input impedance

INTF:T1_FT1:CONFIG:LBO [0|-7.5|-15] :select line buildout

INTF:T1_FT1:CONFIG:REFER_INPUT [TERM|BRIDGE] :select the reference input

INTF:T1_FT1:ERR_INSERT [OFF|BPV|LOGIC|L+BPV|SINGLE|RATE|1FRAME|2FRAME|3FRAME|4FRAME] :select error insertion

INTF:T1_FT1:ERR_RATE (rate string) :program the error insertion rate between 1E-9 to 9E-3

INTF:T1_FT1:FRACT1:TX (number),[ON|OFF] :selects the transmit channel number (1 - 24) and the channel status

INTF:T1_FT1:FRACT1:RX (number),[ON|OFF] :selects the receive channel number (1 - 24) and the channel status

INTF:T1_FT1:FRACT1 [AUTO|CLEAR|64|56] :select the current fractional T1 configuration

T1_FT1 (Continued)

T1/FT1 Interface Control	command/query
INTF:T1_FT1:IDLE:PATTERN (string)	:program the idle code (string =up to eight 1's or 0's)
INTF:T1_FT1:LOOP [CSU|FAC1|FAC2| DL-LLB|DL-PLB|PROGRAM]	:select the loop code
INTF:T1_FT1:LOOP:PROGRAM [UP|DOWN] (string)	:select either a loop-up or loop-down code and program the loop code (string = up to 8 bits of 1's and 0's)
INTF:T1_FT1:MODE [T1|FT1|VOICE| TLB|LLB|ESFDL|RS232]	:select the mode of operation
INTF:T1_FT1:RECEIVE_CHANNEL (number)	:select a receive channel (number = 1-24)
INTF:T1_FT1:RS232:CHANNEL (number),(number)	:select the transmit channel (number = 1-24) and the receive channel (number= 1-24)
INTF:T1_FT1:RS232 [DATALINK| CHANNEL]	:configure the RS232 port
INTF:T1_FT1:VOICE (number),(number)	:select the transmit channel (number = 1 - 24) and the receive channel (number = 1 - 24)
INTF:T1_FT1:VOICE:SIGNAL [ON|OFF]	:enable or disable robbed bit signaling

T1_FT1 (Continued)

T1/FT1 Interface Control	command/query
INTF:T1_FT1:VOICE:SIGNAL: PATTERN (string)	:program the logic state of the ABCD signaling bits
INTF:T1_FT1:CONFIG:CODE?	:report the current coding selection
INTF:T1_FT1:ERR_INSERT?	:report the current type of error insertion
INTF:T1_FT1:ERR_RATE?	:report the current error insertion rate
INTF:T1_FT1:CONFIG:FRAME?	:report the current framing selection
INTF:T1_FT1:CONFIG:INPUT?	:report the current input impedance
INTF:T1_FT1:CONFIG:LBO?	:report the current line buildout
INTF:T1_FT1:IDLE:PATTERN?	:report the current idle pattern
INTF:T1_FT1:CONFIG: REFER_INPUT?	:report the current reference input
INTF:T1_FT1:IDLE:SCAN?	:report which channels contain the idle code
INTF:T1_FT1:FRACT1:TX?	:report the transmit channel number (1 - 24) and the channel status
INTF:T1_FT1:FRACT1:RX?	:report the receive channel number (1 - 24) and the channel status

T1_FT1

T1/FT1 Interface Control	command/query
INTF:T1_FT1:FRACT1?	:report the current fractional T1 configuration
INTF:T1_FT1:LOOP?	:report the current loop code selected
INTF:T1_FT1:LOOP: PROGRAM (string)?	:report the current programmed loop code for either the loop-up or loop-down code (string = UP, DOWN)
INTF:T1_FT1:MODE?	:report the current mode of operation
INTF:T1_FT1: RECEIVE_CHANNEL?	:report the currently selected receive channel
INTF:T1_FT1:RS232: CHANNEL?	:report the transmit channel (number = 1-24) and the receive channel (number= 1-24)
INTF:T1_FT1:RS232?	:report the RS-232 port
INTF:T1_FT1:VOICE?	:report the transmit and receive channels
INTF:T1_FT1:VOICE: SIGNAL?	:report the current robbed bit signaling status
INTF:T1_FT1:VOICE: SIGNAL:PATTERN?	:report the logic state of the ABCD signaling bits

T1_FT1

T1/FT1 Interface Control — command/query

This series of commands and queries enables remote control of the DS1/FT1 Interface.

For specifics on interface settings, please refer to the *DS1/FT1 Interface Manual*.

Example:

>INTF:T1:FT1:VOICE:SIGNAL? :report the current robbed bit signaling status

OFF

>INTF:T1:FT1:VOICE:SIGNAL ON :turn robbed bit signaling on

MIL.449

RS-449/530/MIL Interface Control	command/query
INTF:MIL449:INTERFACE [118114|RS449|EIA530|188C]	:select interface type
INTF:MIL449:MODE [BALANCE|UNBALANCE]	:select mode
INTF:MIL449:BALANCE:IMPEDANCE [78|100|124|UNTERMINATE]	:select the termination impedance
INTF:MIL449:CONNECTOR [TWINAX|DTYPE]	:select connector
INTF:MIL449:CONNECTOR:DTYPE [DTE|DCE_TT|DCE_ST]	:select a D-type connector
INTF:MIL449:EMULATE:SIGNAL [POSITIVE|NEGATIVE]	:select signal polarity
INTF:MIL449:EMULATE:LL [ON|OFF]	:enable or disable LL emulation
INTF:MIL449:EMULATE:RL [ON|OFF]	:enable or disable RL emulation
INTF:MIL449:EMULATE:TM [ON|OFF]	:enable or disable TM emulation
INTF:MIL449:EMULATE:CTS [ON|OFF]	:enable or disable CTS emulation
INTF:MIL449:INTERFACE?	:report the current interface type
INTF:MIL449:MODE?	:report the current mode

MIL449 (Continued)

RS-449/530/MIL Interface Control	command/query
INTF:MIL449:CONNECTOR?	:report the current connector type
INTF:MIL449:CONNECTOR:DTYPE?	:report the current D-type connector
INTF:MIL449:EMULATE:SIGNAL?	:report the current signal emulation
INTF:MIL449:EMULATE:LL?	:report current LL status
INTF:MIL449:EMULATE:RL?	:report current RL emulation status
INTF:MIL449:EMULATE:TM?	:report current TM emulation status

This series of commands and queries enables remote control of the RS-449/530/MIL Interface.

For specifics on interface settings, please refer to the *RS-449/530/MIL Interface Manual*.

Example:

>INTF:MIL449:INTERFACE? :report the current interface type

EIA530

>INTF:MIL449:INTERFACE RS449 :select the RS-449 interface type

DS0A_B

DS0A/B Interface Control	command/query
INTF:DS0A_B:MODE [CLEAR|DS0A|DS0B|MUX_TEST]	:select an interface mode
INTF:DS0A_B:DATA_TYPE [BIPOLAR|LOGIC_NEAR|LOGIC_FAR]	:select a data format
INTF:DS0A_B:LOOP [ALTERNATING|LATCHING]	:select loopback
INTF:DS0A_B:LOOP:ALTERNATING [OCU|OCU+HL96|DSU|CHAN|CHAN+R|CHAN+2R|RPTR1|RPTR2]	:select an alternating loopback
INTF:DS0A_B:LOOP:LATCHING [OCU|CHAN|DS0_DP|LSI|MJU] (number)	:select a latching loopback
INTF:DS0A_B:DS0A [2.4|4.8|9.6|19.2|56]	:select a DS0A data rate
INTF:DS0A_B:DS0B [2.4|4.8|9.6] (number)	:select a DSOB data rate (number = 1-20 for 2.4 kb/s, 1-10 for 4.8 kb/s, and 1-5 for 9.6 kb/s)
INTF:DS0A_B:MUX_TEST [A_B|B_A] [2.4|4.8|9.6] (number)	:select a muxtest data rate and channel number (number = 1-20 for 2.4 kb/s, 1-10 for 4.8 kb/s, and 1-5 for 9.6 kb/s)
INTF:DS0A_B:MJU:SELECT [1|2|3|4]	:select a Multipoint Junction Unit branch
INTF:DS0A_B:MJU:BLOCK [1|2|3|4]	:block the Multipoint Junction Unit branch specified

DS0A_B (Continued)

DS0A/B Interface Control	command/query
INTF:DS0A_B:MJU:UNBLOCK [1|2|3|4]	:unblock the Multipoint Junction Unit branch specified
INTF:DS0A_B:MJU:RELEASE	:release all MJUs in the circuit
INTF:DS0A_B:MJU:RESTORE	:restore the previously selected branches and return them to normal operation
INTF:DS0A_B:THRU_MODE [ON|OFF]	:enable or disable thru-mode
INTF:DS0A_B:TX_BYTE [OFF|SINGLE|REPEAT|BYTE]	:set byte encoder
INTF:DS0A_B:TX_BYTE:DATA (hex)	:enter 8-bit hexidecimal transmit byte code
INTF:DS0A_B:CHANNEL [PRIMARY|SECONDARY]	:select primary or secondary channel testing
INTF:DS0A_B:ERR_INSERT [OFF|DATA|FRAME|DAT+FR|RATE]	:select error insertion
INTF:DS0A_B:ERR_RATE (rate string)	:program the error insertion rate between 1E-9 to 9E-3
INTF:DS0A_B:MODE?	:report the current interface mode
INTF:DS0A_B:DATA_TYPE?	:report the current data format
INTF:DS0A_B:LOOP:ALTERNATING?	:report the current alternating loopback setting

DS0A_B (Continued)

DS0A/B Interface Control	command/query
INTF:DS0A_B:LOOP:LATCHING?	:report the current latching loopback setting
INTF:DS0A_B:DS0A?	:report the current DS0A data rate
INTF:DS0A_B:DS0B?	:report the current DS0B data rate
INTF:DS0A_B:MUX_TEST?	:report the current muxtest data rate
INTF:DS0A_B:THRU_MODE?	:report the current thru-mode status
INTF:DS0A_B:TX_BYTE?	:report the current byte encoder setting
INTF:DS0A_B:TX_BYTE:DATA?	:report the current 8-bit hexidecimal transmit byte code
INTF:DS0A_B:CHANNEL?	:report the current channel (primary or secondary)
INTF:DS0A_B:ERR_INSERT?	:report the current error insertion status
INTF:DS0A_B:ERR_RATE?	:report the current error insertion rate

This series of commands and queries enables remote control of the DS0A/B Interface.

For specifics on interface settings, please refer to the *DS0A/DS0B Interface Manual*.

DS0A_B (Continued)

DS0A/B Interface Control — command/query

Example:

```
>INTF:DS0A_B:THRU_MODE?                    :report the current
                                            thru-mode status
OFF
>INTF:DS0A_B:THRU_MODE ON                  :turn thru-mode on
```

JITTER COMMANDS

The FIREBERD 6000 features a set of commands that configure, operate, and query the status of its jitter generation and measurement capability.

The jitter commands and queries accepted by the FIREBERD 6000 are:

Commands	Queries
JITTER:FILTER	JITTER:FILTER?
	JITTER:GEN?
JITTER:GEN:AMP	JITTER:GEN:AMP?
JITTER:GEN:FREQ	JITTER:GEN:FREQ?
JITTER:GEN:MOD	JITTER:GEN:MOD?
JITTER:HITS_THRESH	JITTER:HITS_THRESH?
JITTER:MASK	JITTER:MASK?
JITTER:RANGE	JITTER:RANGE?
	JITTER:SA?
JITTER:SA:FREQ	JITTER:SA:FREQ?
JITTER:SA:GRAPH	
JITTER:SA:LIST	
JITTER:SA:SWEEP	JITTER:SA:SWEEP?

FILTER

Select DS1 Jitter Bandwidth **command/query**

JITTER:FILTER
[FULL|>8KHZ] :select jitter bandwidth measurement criteria

JITTER:FILTER? :report how jitter is currently measured

The **JITTER:FILTER** command determines how DS1 jitter bandwidth is measured. FULL measures the entire jitter bandwidth; >8KHZ only measures jitter above 8 kHz.

The **JITTER:FILTER?** query reports how DS1 jitter bandwidth is currently measured.

Example:

>JITTER:FILTER? :report the DS1 jitter bandwidth
FULL :full jitter bandwidth is measured
>JITTER:FILTER >8KHZ :set the 8 kHz jitter filter

GEN

Jitter Generation — command/query

JITTER:GEN:MOD [OFF|SINE|SQUARE|TRIANGLE|RAMP|EXT|EXT1|EXT2] :set jitter generator modulation

JITTER:GEN:FREQ [(frequency)|UP|DOWN|SWEEP] :set jitter generator frequency

JITTER:GEN:AMP [(amplitude)|MASK|XFER50%|XFER75%] :set jitter generator amplitude

JITTER:GEN? :report the current jitter modulation, frequency, and amplitude

JITTER:GEN:MOD? :report the current jitter generator modulation

JITTER:GEN:FREQ? :report the current jitter generator frequency

JITTER:GEN:AMP? :report the current jitter generator amplitude

This series of commands and queries is dedicated to jitter generation.

The **JITTER:GEN:MOD** command controls jitter generator modulation. Using this command, these waveforms can be selected:

OFF	:no modulation takes place
SINE	:sine wave
SQUARE	:square wave
TRIANGLE	:triangle wave
RAMP	:ramp wave
EXT	:external waveform (DS1)
EXT1	:external waveform using jitter generator band 1 (G.703)
EXT2	:external waveform using jitter generator band 2 (G.703)

GEN (Continued)

Jitter Generation — command/query

The **JITTER:GEN_FREQ** command controls the jitter generator frequency. (frequency) sets a fixed frequency between 10 Hz and 40 kHz; the fixed frequency is limited to not more than 4 significant digits. UP and DOWN raises or lowers the fixed frequency to the next higher or lower value on the list of 40 valid frequencies. SWEEP sweeps the jitter generator across the same 40 frequencies as the spectrum analyzer, permitting synchronization between both components. See Section 9 of the *FIREBERD 6000 Operations Manual* for detailed information about jitter generator frequencies.

The **JITTER:GEN_AMP** command sets the jitter generator amplitude. The (amplitude) must be in the range of 0.00-13.00 unit intervals (UIs); UP and DOWN increases or decreases the amplitude by 0.02 UIs, respectively. MASK sets the amplitude to the mask currently selected by the JITTER:MASK command; XFER50% and XFER75% sets the amplitude to half or three-fourths of the selected mask.

The **JITTER:GEN?** query reports the current jitter modulation, frequency, and amplitude. The **JITTER:GEN:MOD?**, **JITTER:GEN:FREQ?**, and **JITTER:GEN:AMPLITUDE?** queries individually report jitter modulation, frequency, and amplitude.

Certain combinations of jitter generator, modulation, frequency, and amplitude settings are invalid. If the FIREBERD is configured for an invalid combination of settings, the execution error messsage **Jitter contention** appears in the error buffer, and the jitter generator keeps its last valid values.

To reduce the possibility of an invalid configuration, the FIREBERD delays processing jitter generator commands until all of the commands of the program message have been read. The delayed commands are considered at the same time; an error is flagged only if this final configuration is invalid.

See Section 9 of the *FIREBERD 6000 Operating Manual* for a description of all invalid settings.

GEN (Continued)

Jitter Generation **command/query**

Example:

>JITTER:GEN:AMP 3.5 :set the jitter generator amplitude to 3.5
>JITTER:GEN:MOD OFF :turn jitter generator modulation off
>JITTER:GEN:AMP? :report the current jitter generator amplitude

03.50
>JITTER:GEN:MOD? :report the current jitter generator modulation

OFF
>JITTER:GEN? :report the current jitter generator modulation, frequency, and amplitude

OFF, 10, 03.50

HITS_THRESH

Jitter Hits **command/query**

JITTER:HITS_THRESH [(threshold)|UP|DOWN] :set the jitter hit threshold

JITTER:HITS_THRESH? :report the current jitter hit threshold

The **JITTER:HITS_THRESH** command sets the jitter hit threshold. When testing DS1 jitter, (threshold) is an amplitude ranging from 0.0 to 6.5 UIs, expressed as *n.n*. When testing G.703 jitter, the value of (threshold) is based on the band (i.e., BAND1 or BAND2) and the interface installed, (64 kb/s or 2048 kb/s), as follows:

Interface	Band	Range	Increments
2048 kb/s	Band 1	0.1 - 6.0 UIp	.1 UIp
	Band 2	0.05 - 0.60 UIp	.05 UIp
64 kb/s	Band 1	0.05 - 0.75 UIp	.05 UIp
	Band 2	0.05 - 0.15 UIp	.05 UIp

The **JITTER:HITS_THRESH?** query reports the current jitter hit threshold.

UP or DOWN can be used to respectively raise or lower the peak jitter threshold by 0.1 UI.

Example:

>JITTER:HITS_THRESH 0.10 :set the jitter hit threshold to 0.10
>JITTER:HITS_THRESH? :report the current jitter hit threshold
0.10

MASK

Jitter Masks **command/query**

JITTER:MASK ["O.171"|"41451"
|"62411-1983"|"62411-1985"|
"43801"|"G.823"|"G.823n"|
"D.PTT"] :select a jitter mask

JITTER:MASK? :report the currently selected jitter mask

The **JITTER:MASK** command selects masks which reflect specific T1 or G.703 jitter standards. The T1 jitter masks which can be selected reflect the standards set forth in CCITT Recommendation O.171, and Bell Technical Reference Publications PUB41451, PUB43801, and PUB62411 (1983 and 1985 issues). The G.703 jitter masks which can be selected reflect the standards set forth in CCITT Recommendations O.171 and G.823, as well as DUTCH PTT standards.

The **JITTER:MASK?** query reports the currently selected jitter mask.

Example:

>JITTER:MASK "O.171" :select the O.171 jitter mask
>JITTER:MASK? :report the currently selected jitter mask

"O.171"

RANGE

G.703 Jitter Measurement Range — command/query

JITTER:RANGE [BAND1|BAND2] :select the measurement range

JITTER:MASK? :report the currently selected measurement range (BAND1 or BAND2)

The **JITTER:RANGE** command determines the range of G.703 jitter bandwidth that is measured during any given test. The range is based on the BAND selected and the interface installed, as follows:

2048 kb/s Interface	• BAND1 — Up to 12.00 UI p-p. • BAND2 — Up tp 1.200 UI p-p.
64 kb/s Interface	• BAND1 — Up to 1.500 UI p-p. • BAND2 — Up tp 0.300 UI p-p.

The **JITTER:RANGE?** query reports how wideband jitter bandwidth is currently measured.

Example:

>JITTER:RANGE? :report the currently selected range band
BAND2
>JITTER:RANGE BAND1 :select BAND1

SA

Jitter Spectrum Analyzer **command/query**

JITTER:SA:SWEEP [OFF|ON] :set spectrum analyzer sweep

JITTER:SA:FREQ [(frequency)|UP|DOWN] :set jitter spectrum analyzer frequency

JITTER:SA:LIST :send a list of spectrum analyzer results to the printer buffer

JITTER:SA:GRAPH :graph spectrum analyzer results

JITTER:SA? :report the current jitter spectrum analyzer sweep and frequency settings

JITTER:SA:SWEEP? :report the current state of the jitter spectrum analyzer sweep

JITTER:SA:FREQ? :report the current jitter spectrum analyzer frequency

This series of commands and queries is dedicated to the FIREBERD 6000's jitter spectrum analyzer.

The **JITTER:SA:SWEEP** command controls the spectrum analyzer sweep. ON sweeps jitter through all 40 frequencies; OFF turns the spectrum analyzer sweep off. When using any T1 interface with jitter capability, the spectrum analyzer sweeps these frequencies:

10, 20, 30, 40, 50, 60, 70, 80,
100, 133, 166, 200, 233, 300, 400, 500, 600, 700, 800,
1000, 1330, 1660, 2000, 2330, 3000, 4000, 5000, 6000, 7000, 8000,
10000, 13300, 16600, 20000, 23300, 26600, 30000, 33300, 36600, and 40000 Hz.

SA (Continued)

Jitter Spectrum Analyzer — command/query

When using the 64 kb/s G.703 Interface, the spectrum analyzer sweeps these frequencies:

10, 20, 30, 40, 50, 60, 70, 80, 100, 133, 166, 200, 233, 300, 400, 500, 600, 700, 800,1000, 1330, 1660, 2000, 2330, 3000, 4000, 5000, 6000, 7000, 8000, 10000, 13300, 16600, 20000, 23300, 26600, and 30000 Hz.

When using the 2048 kb/s G.704 Interface, the spectrum analyzer sweeps these frequencies:

10, 20, 30, 40, 50, 60, 70, 80, 100, 133, 166, 200, 233, 300, 400, 500, 600, 700, 800,1000, 1330, 1660, 2000, 2330, 3000, 4000, 5000, 6000, 7000, 8000, 10000, 13300, 16600, 20000, 23300, 26600, 30000, 36600, 40000, 43300, 46600, 50000, 53300, 56600, 60000, 63300, 66000, 70000, 73300, 76600, 80000, 83300, 86600, 90000, 93300, 96600, and 100000 Hz.

When the jitter spectrum analyzer sweep is turned OFF, the **JITTER:SA:FREQ** command is used to select a single frequency. (frequency) is the frequency to be monitored for jitter: UP or DOWN changes the current selection to the next highest or lowest frequency based on the interface that is selected.

The **JITTER:SA:LIST** command initiates a printout of jitter spectrum analyzer results.

The **JITTER:SA:GRAPH** command initiates a jitter spectrum analyzer frequency graph.

The **JITTER:SA?** query reports the current jitter spectrum analyzer and frequency settings.

The **JITTER:SA:SWEEP?** query reports the current state of the jitter spectrum analyzer sweep (i.e., ON or OFF).

The **JITTER:SA:FREQ?** query reports the current jitter spectrum analyzer frequency.

SA (Continued)

Jitter Spectrum Analyzer **command/query**

Example:

>JITTER:SA:SWEEP?	:report the current state of the spectrum analyzer sweep
OFF	
>JITTER:SA:SWEEP ON	:set the spectrum analyzer to sweep all frequencies
>JITTER:SA:SWEEP OFF	:turn the spectrum analyzer sweep off
>JITTER:SA:FREQ 1330	:monitor jitter at 1330 Hz (valid frequency for T1 interfaces)
>JITTER:SA?	:report the current spectrum analyzer sweep and frequency settings
OFF, 1330	

PRINT COMMANDS

PRINT commands control the FIREBERD 6000's print capability. These commands can be used to obtain hard copy test results or a printout of the FIREBERD's current configuration.

The print commands and queries are:

Commands	Queries
PRINT:CONTROLS	PRINT:CONTROLS?
PRINT:DEVICE	PRINT:DEVICE?
PRINT:EVENT	PRINT:EVENT?
	PRINT:EVENT:NORMAL?
PRINT:EVENT:NORMAL:ERROR	PRINT:EVENT:NORMAL:ERROR?
PRINT:EVENT:NORMAL:SYNC_LOSS	PRINT:EVENT:NORMAL:SYNC_LOSS?
PRINT:EVENT:NORMAL:TEST_INT	PRINT:EVENT:NORMAL:TEST_INT?
PRINT:EVENT:NORMAL:TIMED	PRINT:EVENT:NORMAL:TIMED?
PRINT:HISTOGRAM	
PRINT:HISTOGRAM:AUTO	PRINT:HISTOGRAM:AUTO?
PRINT:HISTOGRAM:FORMAT	PRINT:HISTOGRAM:FORMAT?
PRINT:HISTOGRAM:PERIOD	PRINT:HISTOGRAM:PERIOD?
PRINT:HISTOGRAM:RESULT	PRINT:HISTOGRAM:RESULT?
PRINT:HISTOGRAM:SAMPLES	PRINT:HISTOGRAM:SAMPLES?
PRINT:HISTOGRAM:VSCALE	PRINT:HISTOGRAM:VSCALE?
PRINT:MODE	PRINT:MODE?
PRINT:PRINTER	PRINT:PRINTER?
PRINT:RESULTS	PRINT:RESULTS?
PRINT:RESULTS:CUSTOM	PRINT:RESULTS:CUSTOM?
PRINT:RESULTS:TYPE	PRINT:RESULTS:TYPE?
PRINT:STATUS	PRINT:STATUS?
PRINT:TERM	PRINT:TERM?
PRINT:WIDTH	PRINT:WIDTH?

CONTROLS

Controls Printouts **command/query**

PRINT:CONTROLS? :send a controls printout to the remote control output queue

PRINT:CONTROLS :generate a controls printout

The **PRINT:CONTROLS?** query sends a controls printout to the remote control output queue. A controls print lists the current state of all FIREBERD 6000 mainframe controls. When applicable, the state of interface controls are also listed.

The **PRINT:CONTROLS** command generates a controls printout. When specified, this command obtains the controls print from the remote control output queue.

Example:

>PRINT:CONTROLS? :send a controls printout to the remote control output queue

. :the controls printout is generated on your printer

>PRINT:CONTROLS :obtain the controls printout from the remote control output queue

DEVICE

Printer Device Selection **command/query**

PRINT:DEVICE
[NONE|232|488|REMOTE] :select a printer output device

PRINT:DEVICE? :report the current printer output selection

The **PRINT:DEVICE** command is used to select a printer output device. 232, 488, and REMOTE select a print device at an RS-232, IEEE-488, or remote output, respectively. NONE indicates that none of the three printer output devices are selected. Note that a printer and a remote control device cannot share the same output.

The **PRINT:DEVICE?** query reports the current printer output device selection.

Example:

>PRINT:DEVICE? :report the current printer output device selection

232
>PRINT:DEVICE NONE :change the current printer output device selection to none

EVENT

Print Event Selection	command/query
PRINT:EVENT [NORMAL|DELTA]	:select normal or delta printouts
PRINT:EVENT:NORMAL: ERROR (error),[OFF|ON]	:print on the occurrence of a specific error
PRINT:EVENT:NORMAL: TIMED [OFF|(minutes)]	:print on a timed interval
PRINT:EVENT:NORMAL: SYNC_LOSS [OFF|ON]	:print on a synchronization loss
PRINT:EVENT:NORMAL: TEST_INT [OFF|ON]	:print on a test interval
PRINT:EVENT?	:report the current printout selection (NORMAL or DELTA)
PRINT:EVENT:NORMAL?	:report all currently selected normal print events
PRINT:EVENT:NORMAL: ERROR? (error)	:report the current print event status of any error (enabled or disabled)
PRINT:EVENT:NORMAL: TIMED?	:report the current timed print event setting
PRINT:EVENT:NORMAL: SYNC_LOSS?	:report the current synchronization loss print event setting
PRINT:EVENT:NORMAL: TEST_INT?	:report the current test interval print event setting

EVENT (Continued)

Print Event Selection — command/query

These commands and queries are dedicated to *print events*. Print events determine when the FIREBERD 6000 automatically generates a results printout.

The **PRINT:EVENT** command selects normal or delta printouts. NORMAL provides cumulative results based on one or more specific events: a time interval, an error, a synchronization loss, and a test interval. DELTA only provides those results accumulated during the most recent test interval.

The **PRINT:EVENT:NORMAL:ERROR** command generates a printout on the occurrence of errors. (errors) may be one or more of the following:

BIT_ERRS	:bit errors
BLK_ERRS	:block errors
BPVS	:bipolar violations
CODE_ERR	:code error
CRC_ERR	:cylic redundancy check error
ERR-SES	:severely errored seconds
FAS_ERR	:frame alignment signal error
FRA_ERR	:frame error
JTR_HITS	:jitter hits
PAT_SLIP	:pattern slip

ON and OFF enable or disable the print error event activity.

The **PRINT:EVENT:NORMAL:TIMED** command allows a printout to be generated at a time interval. (minutes) can be any value within the range 1-1440; ON and OFF enable or disable the timed print event activity.

The **PRINT:EVENT:NORMAL:SYNC_LOSS** command permits a printout to be generated when pattern synchronization is lost; in such instances, the results that are sent to the printer are those accumulated from 2 to 4 seconds preceding the synchronization loss. ON and OFF enable or disable the synchronization loss print event activity.

EVENT (Continued)

Print Event Selection — command/query

The **PRINT:EVENT:NORMAL:TEST_INT** command permits a printout to be generated at the end of the current test interval. Test interval settings are controlled by the CONFIG:TEST_INT command. ON and OFF turn the test interval print event activity on and off, respectively.

The **PRINT:EVENT?** query reports the current printout selection (NORMAL or DELTA).

The **PRINT:EVENT:NORMAL?** query reports the status of all normal print events. Since the response to this query may change in size in the future, caution is advised when issuing this query.

The **PRINT:EVENT:NORMAL:ERROR?** query reports the current error print event settings for a specific error.

The **PRINT:EVENT:NORMAL:TIMED?** query reports the current timed print event setting.

The **PRINT:EVENT:NORMAL:SYNC_LOSS?** query reports the current synchronization loss print event setting.

The **PRINT:EVENT:NORMAL:TEST_INT?** query reports the current test interval print event setting.

EVENT (Continued)

Print Event Selection **command/query**

Example:

>PRINT:EVENT? :report the current print event

DELTA

>PRINT:EVENT NORMAL :set the print event to normal

>PRINT:EVENT:NORMAL:TIMED 60?

:generate a printout every hour

>CONFIG:TEST_INT? :report the current test interval

"1.0E6"

>CONFIG:TEST_INT "1.0E4" :printouts are generated at the completion of the 1.0E4 test interval

>PRINT:EVENT:NORMAL:ERROR? BPVS

:report the status of print events caused by BPVs

OFF :no printouts are generated during the occurrence of BPVs

>PRINT:EVENT:NORMAL:ERROR BPVS,ON

:enable printouts during the occurrence of BPVs

HISTOGRAM

Histogram Setup — command/query

PRINT:HISTOGRAM:AUTO [ON|OFF] :enable or disable results histogram printouts

PRINT:HISTOGRAM:RESULT (1,2,3),[OFF|(result)] :select the result to be histogrammed and assign it a number (1-3)

PRINT:HISTOGRAM [1,2,3] :print any or all results histograms

PRINT:HISTOGRAM:VSCALE (1,2,3),(scale) :select the vertical scale for the histogram number specified

PRINT:HISTOGRAM:SAMPLES [5-60] :select the number of samples for automatic histograms

PRINT:HISTOGRAM:PERIOD [1MIN|15MIN|1HOUR|1DAY] :select a sampling period

PRINT:HISTOGRAM:FORMAT [GRAPH|LIST] :select a histogram format

PRINT:HISTOGRAM:AUTO? :report the current status of histogram printouts (enabled or disabled)

PRINT:HISTOGRAM:RESULT? (1,2,3) :report the current result to be histogrammed by number (1-3)

HISTOGRAM

Histogram Setup **command/query**

PRINT:HISTOGRAM:VSCALE?(1,2,3) :report the current vertical scale for the histogram number specified (1-3)

PRINT:HISTOGRAM:SAMPLES? :report the current number of samples for automatic histograms

PRINT:HISTOGRAM:PERIOD? :report the current sampling period

PRINT:HISTOGRAM:FORMAT? :report the current histogram format

This series of commands and queries is dedicated to *histograms*. Histograms offer a way to measure how a FIREBERD 6000 test result changes with time. For example, a histogram can be used to monitor bit errors that have accumulated in each hour over a 60-hour period.

The **PRINT:HISTOGRAM:AUTO** command turns the histogram activity ON or OFF.

The **PRINT:HISTOGRAM** command allows the printing of simultaneous histograms for up to 3 test results.

The **PRINT:HISTOGRAM:RESULT** command allows the selection of up to 3 results by number (1-3). The (result) may be any of the following:

1SEC_CRC	:one-second CRC errors
1S_DEN_S	:ones density violation seconds
AIS_SEC	:alarm indication signal
AVL_SEC	:available seconds
BIT_ERRS	:bit errors
BLK_ERRS	:block errors

HISTOGRAM (Continued)

Histogram Setup **command/query**

BLOCKS	:block errors
BPVS	:bipolar violations
C-D_CHA	:clock-data phase changes
CHAR_ERR	:asynchronous character errors
CLK_LOSS	:clock loss
CODE_ERR	:code error
CRC_ERR	:cyclic redundancy check error
DAT_LOSS	:data loss
DEG_MIN	:degraded minutes
EA_SEC	:errored analysis seconds
EF_EAS	:error-free error analysis seconds
ERR_SES	:bit errors minus severely errored seconds
ERR-EAS	:errored error analysis seconds
FAS_ERR	:frame aligment signal error
FCS_ERR	:frame check sequence errors
FRA_ERR	:frame error
FRA_LOSS	:frame loss
FRMR	:frame rejection frames
GERR_SEC	:G.821 errored second
G_EFS	:G.821 error-free seconds
G_VFS	:G.821 violation-free seconds
L2_ABORT	:layer 2 frame aborts
L2_FRMS	:number of layer 2 frames
L2_F_ERR	:layer 2 frame errors
PAT_LOSS	:pattern synchronization loss
PAT_SLIP	:pattern slip
PATL_SEC	:pattern loss seconds
REJ_FRM	:rejection frames
REM_SEC	:remote seconds
SES	:severely errored seconds
SIG_LOSS	:signal loss
SIGL_SEC	:signal loss seconds

HISTOGRAM (Continued)

Histogram Setup command/query

SVS	:severely violated seconds
UNA_SEC	:unavailable seconds
VF_EAS	:violation-free error analysis seconds
V_FR_SEC	:violated frame seconds
XSO_SEC	:excess zero seconds
YEL_SEC	:yellow alarm seconds

PRINT:HISTOGRAM:RESULT OFF disables any of the histogram results listed.

The **PRINT:HISTOGRAM:VSCALE** command selects the vertical scale for the histogram number (1-3) specified. The (scale) can be any of the following:

10, 20, 50, 100, 200, 500, 1K, 2K, 5K, 10K, 20K, 50K, 100K, 200K, 500K, 1M, 2M, 5M, 10M, 20M, 50M, AUTO

The **PRINT:HISTOGRAM:SAMPLES** command selects the number of samples required for all 3 histograms. Any number within the range 5-60 can be specified.

The **PRINT:HISTOGRAM:PERIOD** command selects the period between result samples for all 3 histograms. Valid periods are 1 minute (1MIN), 15 minutes (15MIN), 1 hour (1HOUR), and 1 day (1DAY).

The **PRINT:HISTOGRAM:FORMAT** command selects a graph or list histogram format. See Section 7 of the *FIREBERD 6000 Operating Manual* for a sample of both.

HISTOGRAM (Continued)

Histogram Setup — command/query

The **PRINT:HISTOGRAM:AUTO?** query reports the current status of histogram printouts (enabled or disabled).

The **PRINT:HISTOGRAM:RESULT?** query reports the current results that are histogrammed.

The **PRINT:HISTOGRAM:VSCALE?** query reports the currently selected vertical scale.

The **PRINT:HISTOGRAM:SAMPLES?** query reports the current number of requested samples.

The **PRINT:HISTOGRAM:PERIOD?** query reports the current sampling period.

The **PRINT:HISTOGRAM:FORMAT?** query reports the current histogram format.

Example:

>PRINT:HISTOGRAM:RESULT 1,BIT_ERRS
:assign bit errors as histogram 1

>PRINT:HISTOGRAM:PERIOD?
:report the current histogram period

15MIN

>PRINT:HISTOGRAM:PERIOD 1DAY
:the new histogram period is 1 day

>PRINT:HISTOGRAM:VSCALE? 1
:report the current vertical scale for histogram 1

100

HISTOGRAM (Continued)

Histogram Setup **command/query**

Example: (Continued)

```
>PRINT:HISTOGRAM:VSCALE 1,200
                                        :the new vertical scale is
                                         200
>PRINT:HISTOGRAM:FORMAT GRAPH
                                        :select a histogram graph
>PRINT:HISTOGRAM 1                      :print histogram 1
HISTOGRAM GRAPH
.
.
.                                       :a bit error histogram
                                         graph is printed
```

MODE

Select Fast or Slow Print Mode **command/query**

PRINT:MODE [FAST|SLOW] :set the printer mode

PRINT:MODE? :report the current print mode

The **PRINT:MODE** command sets print speed. FAST enables printing as fast as the FIREBERD can print; SLOW sets the print speed to no more than 20 characters per second with an additional one-half second delay at the end of each printed line.

The **PRINT:MODE?** query reports the current print mode setting.

Example:

>PRINT:MODE? :report the current print mode setting

SLOW

>PRINT:MODE FAST :change the print mode to fast

PRINTER

Enable/Disable Printer Output **command/query**

PRINT:PRINTER
[ON|OFF] :enable or disable printer output

PRINT:PRINTER? :report the current state of printer output

The **PRINT:PRINTER** command controls the overall print function. ON enables printer output; OFF disables all print functions and clears the print buffer. When OFF is selected, no printouts are generated except those that are specifically requested via the PRINT:RESULTS or PRINT:CONTROLS commands.

The **PRINT:PRINTER?** query reports the current state of printer output.

Example:

>PRINT:PRINTER? :report the current print mode setting
OFF
>PRINT:PRINTER ON :enable printer output

RESULTS

Results Printouts **command/query**

PRINT:RESULTS? :send a results printout to the remote control output queue

PRINT:RESULTS :obtain a results printout (i.e., release the results printout from the queue)

PRINT:RESULTS:TYPE [STD|LONG|CUSTOM] :select a results printout format

PRINT:RESULTS:CUSTOM [(result)|ALL] [OFF|ON] :include/exclude a specific result from a customized printout or include/exclude all results

PRINT:RESULTS:TYPE? :report the current results printout format

PRINT:RESULTS:CUSTOM? :report the current state of any result

This series of commands and queries are dedicated to results printouts. A results print is a hard copy of the results accumulated by the FIREBERD 6000. If desired, they can be customized to include a specific set of test results.

The **PRINT:RESULTS?** query sends a controls printout to the remote control output queue.

The **PRINT:RESULTS** command prints the controls printout sent to the remote control output queue via the **PRINT RESULTS?** query.

The **PRINT:RESULTS:TYPE** command selects standard, long, or custom results prints. STD prints an abbreviated list of key results; LONG prints all available results; CUSTOM customizes a printout to include any combination of test results. Before enabling custom printouts, use the **PRINT:RESULT:CUSTOM** command to do the actual customizing.

RESULTS (Continued)

Results Printouts — command/query

The **PRINT:RESULTS:TYPE:CUSTOM** command is used to include or exclude a specific result from a customized printout, or include/exclude all results from the customized printout. ON includes results; OFF excludes results. After customizing the printout, use the **PRINT:RESULTS:TYPE** command to enable custom printing.

The (result) names which can be customized may be any of the following:

Error Results

AVG_BER	:average bit error rate
AVG_BLER	:average block error rate
BIT_ERRS	:bit errors
BER	:bit error rate
BLOCKS	:blocks
BLK_ERRS	:block errors
CHAR_ERR	:asynchronous character errors
PAT_SLIP	:pattern slip

Performance Results

%AVL_SEC	:percent available seconds
%DEG_MIN	:percent degraded minutes
%SES	:percent severely errored seconds
%SVS	:percent severely violated seconds
AVL_SEC	:available seconds
BER_SES	:bit error rate minus severely errored seconds
DEG_MIN	:degraded minutes
ERR_SES	:bit errors minus severely errored seconds
G_%EFS	:G.821 percent error-free seconds
G_EFS	:G.821 error-free seconds

RESULTS (Continued)

Results Printouts **command/query**

Performance Results (Continued)

GERR_SEC	:G.821 errored seconds
G_%VFS	:percent violation-free seconds
G_VFS	:violation-free seconds
SES	:severely errored seconds
SVS	:severely violated seconds
UNA_SEC	:unavailable seconds

Time Results

%ACT_SEC	:percent active seconds
%PATL_SEC	:percent pattern loss seconds
DATE	:date
EA_SEC	:errored error analysis seconds
EF_EAS	:error-free error analysis seconds
ERR_EAS	:errored error analysis seconds
ELAP_SEC	:elapsed seconds
PATL_SEC	:pattern loss seconds
SIGL_SEC	:signal loss seconds
TIME	:time

Signal Results

15m_WNDR	:15-minute wander
24h_WNDR	:24-hour wander
+_LVL_dB	:positive level in decibels
+_LVL_V	:positive level in volts
-_LVL_dB	:negative level in decibels
-_LVL_V	:negative level in volts
+WNDR	:positive wander
-WNDR	:negative wander
%MASK	:maximum percent of jitter mask
1SEC_JTR	:one-second jitter
DAT_RATE	:data rate
DELAY	:delay

RESULTS (Continued)

Results Printouts **command/query**

Signal Results (Continued)

GEN_FREQ	:generator frequency
JTR_HITS	:jitter hits
LVL_dBm	:level of all-ones signal in dBm
LVL_V	:level of all-ones signal in volts
MAX_JTR	:maximum jitter
PP_LVL_V	:peak-to-peak level in volts
PP_WNDR	:peak-to-peak wander
PWR_STAT	:ISDN power feeding status
RCV_BYTE	:receive byte
RCV_CODE	:receive code
RCV_FREQ	:receive frequency
R_LVL_dB	:receiver signal level in decibels
R_LVL_V	:receiver signal level in volts
RX_INFO	:receiver information sequence state
SMPX_CUR	:simplex current
TEI	:terminal equipment identifier assigned
TIE_WNDR	:tie wander
TX_INFO	:transmitter information sequence state

T-CARRIER Results

%VF_EAS	:percent violation-free error analysis seconds
1SEC_CRC	:one-second CRC errors
AVG_BPVR	:average bipolar violation rate
AVG_CER	:average code error rate
AVG_CRC	:average CRC errors
AVG_FAS	:average frame alignment signal error rate
AVG_FER	:average frame error rate
BIT_SLIP	:bit slips
BPV_RATE	:bipolar violation rate
BPVs	:bipolar violations
CER	:code error rate

T-Carrier Results (Continued)

CODE_ERR	:code errors
CRC_ERR	:CRC errors
CRC_E_RT	:CRC error rate
FAS_ERR	:frame alignment signal error
FAS_E_RT	:frame alignment signal error rate
FCS_ERR	:frame check sequence errors
FE_RATE	:frame error rate
FRA_ERR	:frame error
FRA_LOSS	:frame synchronization loss
FRMR	:frame rejection frames
L2_ABORT	:layer 2 frame abort
L2_F_ERR	:layer 2 frame errors
L2_FRMS	:number of layer 2 frames
MAX_ZERO	:maximum consecutive zeros
RCV_BOM	:received bit-oriented message
REJ_FRM	:rejection frames
RX_ABCD	:DS0 ABCD Signal
VF_EAS	:violation-free error analysis seconds

Alarm Results

1S_DEN_S	:ones density violation seconds
AIS_SEC	:alarm indication signal seconds
C-D_CHA	:clock-data phase change
CLK_LOSS	:receiver clock loss count
DAT_LOSS	:data loss
PAT_LOSS	:pattern synchronization loss count
PWR_LOSS	:power loss
REM_SEC	:remote alarm seconds
SIG_LOSS	:receiver signal loss
V_FR_SEC	:violated frame seconds
YEL_SEC	:yellow alarm seconds
XS0_SEC	:excess zero seconds

RESULTS (Continued)

Results Printouts — command/query

The **PRINT:RESULTS:TYPE?** query reports the current results printout format.

The **PRINT:RESULTS:CUSTOM?** report the current state of any test result (i.e., included or excluded).

Example:

```
>PRINT:RESULTS_TYPE LONG      :select long results printout
>PRINT:RESULTS?               :queue the long results printout
.
.
.
.                             :a long results print is sent to your
                               terminal or computer
>PRINT:RESULTS                :print the long results printout
```

STATUS

Status Printouts **command/query**

PRINT:STATUS (type),[OFF|ON] :include/exclude a specific status print

PRINT:STATUS? (type) :report the current state of status printouts

The **PRINT:STATUS** command controls which of the FIREBERD 6000's status prints are enabled. The (type) can be any of the following:

ASYNC_FRAME_ERROR	:asynchronous frame error
CLOCK_LOSS	:clock loss
DATA_LOSS	:data loss
FRAME_SYNC_LOSS	:frame sync loss
G.821_TRANSITION	:G.821 transition
INTERFACE	:interface
RCVR_CLK_POL_CHNG	:receiver clock polarity change
POWER_UP	:power up
SIGNALING_CHANGE	:frame sync loss
SIGNAL_LOSS	:signal loss
SYNC_ACQUIRED	:sync acquired
SYNC_LOSS	:sync loss
TEST_RESTART	:test restart

Note that status printouts are only generated when the PRINT:PRINTER command is ON (i.e., the FIREBERD 6000's print capability is enabled).

The **PRINT:STATUS?** query reports the current state of status printouts.

STATUS (Continued)

Status Printouts — command/query

Example:

```
>PRINT:STATUS? SYNC LOSS          :report the current state of status
                                   printouts when synchronization
                                   is lost
ON
>PRINT:STATUS? DATA LOSS, OFF     :disable status prints when data
                                   is lost
```

TERM

Line Terminator **command/query**

PRINT:TERM
[CR|LF|CRLF] :set the line terminator

PRINT:TERM? :report the currently selected line terminator

The **PRINT:TERM** command sets the line terminator sequence which is transmitted at the end of each printed line. Valid line terminators are carriage return (CR), line feed (LF), and carriage return/line feed pair (CRLF).

The **PRINT:TERM?** query reports the current line terminator.

Example:

>PRINT:TERM LF :set the line terminator to line feed
>PRINT:TERM? :report the current line terminator
LF

WIDTH

Printout Width **command/query**

PRINT:WIDTH
[20|40|80] :set the number of columns per line

PRINT:WIDTH? :report the current printer width

The **PRINT:WIDTH** command sets the printer line width; 20, 40, or 80 columns can be specified. Note that the width should be set in accordance with the size of the printer or CRT being used.

Example:

>PRINT:WIDTH 40 :set width to 40 columns
>PRINT:WIDTH? :report the current line width
40

REMOTE CONTROL COMMANDS

REMOTE CONTROL commands are used to set FIREBERD 6000 functions that are not grouped with other commands.

The REMOTE CONTROL commands and queries are:

Commands	Queries
LOCAL	
LOCKOUT	LOCKED?
RC_MODE	RC_MODE?
UNLOCK	

LOCAL

Local Mode **command**

LOCAL :enter local mode

The **LOCAL** command returns the FIREBERD 6000 to local mode and makes the front-panel switches active. The FIREBERD remains in local mode until it receives a valid remote control command from the controller.

This command is not valid when using the IEEE-488 interface. When the IEEE-488 interface is used, the Go-To-Local (GTL) bus-level command performs a similar function. See your controller's documentation.

Example:

>LOCAL :enter local (front panel) mode

LOCKOUT

Front-Panel Lockout (RS-232 Only) **command**

LOCKOUT :lock out the front panel

UNLOCK :unlock the front panel

LOCKED? :report whether the front panel is locked out

The **LOCKOUT** command disables all FIREBERD 6000 front-panel control so that it is not possible to affect a test from the front panel.

The **UNLOCK** command enables the ENTER key on the FIREBERD 6000's MENU keypad. When pressed, the ENTER key returns the FIREBERD to front-panel control. This command does not directly enable all front-panel controls. See the LOCAL command.

The **LOCKED?** query displays the current status of this command A value of 1 indicates the front panel is completely disabled; a value of 0 indicates that the ENTER key can be pressed to restore front-panel operation.

These commands are not valid when using the IEEE-488 interface. When the IEEE-488 interface is used, the bus-level local-lockout (LOL) command performs a similar function. See your controller's documentation.

Example:

>LOCKOUT :the front panel is disabled
>LOCKED? :check the locked state
1
>UNLOCK :press the ENTER key on the front panel to restore front-panel operation

RC_MODE

Remote Control Mode **command/query**

RC_MODE [TERMINAL|COMPUTER] :place the FIREBERD in terminal or computer mode

RC_MODE? :return the current remote control mode

When using the RS-232 Remote Control Interface, the **RC_MODE** command places the FIREBERD 6000 in either terminal or computer remote control mode. Computer mode is specified when using a programmed controller; terminal mode is specified when using a dumb terminal. See Section 2 for additional information on both terminal and computer modes.

The **RC_MODE?** query reports which mode of remote control operation the FIREBERD is configured for.

Example:

>RC_MODE? :return the remote control mode
TERMINAL

>RC_MODE COMPUTER :place the FIREBERD 6000 in computer mode

RESULT

The RESULT commands and queries are used to retrieve test results. In addition to the basic ability to query individual measurements, it is also possible to capture all of the results that are available in a single test.

The RESULTS commands and queries are:

Commands	Queries
RESULT:CLEAR	
	RESULT?
RESULT:DISPLAY:HOLD	RESULT:DISPLAY:HOLD?
RESULT:DISPLAY	RESULT:DISPLAY?
	RESULT:HELD?
RESULT:DISPLAY:PRE_SYNCLOSS	
RESULT:HOLD	
RESULT:PRE_SYNCLOSS	
RESULT:RESTART	

CLEAR

Clear Result **command**

RESULT:CLEAR :clear all test result values

The **RESULT:CLEAR** command clears all test result values without forcing the receiver to resynchronize. This command does not cause a test restart.

Example:

>RESULT:CLEAR :clear all test result values

DISPLAY

Control Results Display **command/query**

RESULT:DISPLAY [1|2] :show the specified (result name) in the left or right results display (1 or 2, respectively)

(result_name)

RESULT:DISPLAY? [1|2] :return the name of the result in the left or right results display

The **RESULT:DISPLAY** command shows the (result_name) and its current value in the front-panel results display specified (1 or 2). The **RESULT** command offers a list of all valid result names.

The **RESULT:DISPLAY?** query returns the name of the result displayed in the left and right windows, respectively.

Example:

>RESULT:DISPLAY? 1 :return the result in the left results display

BIT_ERRS
>RESULT:DISPLAY 1,BER :display the current BER in the left results display

DISPLAY:HOLD

Control Display Hold **command/query**

RESULT:DISPLAY:HOLD [ON|OFF] :turn display hold ON or OFF

RESULT:DISPLAY:HOLD? :return the current state of the display hold activity

The **RESULT:DISPLAY:HOLD** command controls the FIREBERD 6000's display hold activity. ON enables display hold; OFF turns display hold off. This command is identical in function to the DISPLAY HOLD switch, located on the FIREBERD's front panel.

The **RESULT:DISPLAY:HOLD?** query returns the current state of the display hold function.

Example:

>RESULT:DISPLAY:HOLD? :report the current setting
OFF
>RESULT:DISPLAY:HOLD ON :enable display hold

DISPLAY:PRE_SYNCLOSS

View Pre-Sync Loss Results at Front Panel **command**

RESULT:DISPLAY:
PRE_SYNCLOSS :display pre-sync loss test results on the FIREBERD front panel

The **RESULT:DISPLAY:PRE_SYNCLOSS** command enables pre-sync loss test results to be displayed on the FIREBERD front panel. This remote control command automatically initiates a **RESULT:DISPLAY:HOLD** command so that these results may be examined for an extended period of time.

Example:

>RESULT:DISPLAY:PRE_SYNCLOSS :recall pre-sync loss results; a pre-sync loss results listing is displayed

>RESULT:DISPLAY:HOLD OFF :release the automatically generated display hold

HELD

Read Held Results **query**

RESULT:HELD?
(result_name) :read the results that are held in the remote control RESULTS HOLD buffer

The **RESULT:HELD?** query allows the user to read the test results that are stored in the RESULTS HOLD buffer. The storing of test results is invoked by the **RESULT:HOLD** command.

Example:

```
>RESULT:HOLD                  :store all test results in the RESULTS
                               HOLD buffer
>RESULT:HELD? BIT_ERRS        :read the number of bit errors that were
                               counted when the RESULT:HOLD
                               command was specified
>RESULT? BIT_ERRS             :show the current bit error count
2
```

HOLD

Result Hold **command**

RESULT:HOLD
[ON|OFF] :hold or release current test results in the RESULTS HOLD buffer

The **RESULT:HOLD** command stores all current test results in the RESULTS HOLD buffer. This allows the user to read test results without their values changing. (See also: **RESULT:HELD**.)

Example:

>RESULT:HOLD :store all test results in the remote control RESULTS HOLD buffer

PRE_SYNCLOSS

Pre-Sync Loss Results at Remote Controller command

RESULT:PRE_SYNCLOSS :display pre-sync loss test results at the remote controller

The **RESULT:PRE_SYNCLOSS** command enables pre-sync loss test results to be displayed at the remote controller. This remote control command automatically initiates a **RESULT:HOLD** command so that these results may be examined for an extended period of time.

Example:

>RESULT:PRE_SYNCLOSS :recall pre-sync loss results
. :a pre-sync loss results listing is displayed at the remote controller
.
.
>RESULT:HOLD OFF :release the display hold which is automatically generated by the RESULT:PRE_SYNCLOSS command

RESTART

Test Restart **command**

RESULT:RESTART :restart a test

The **RESULT:RESTART** command clears all test results and alarms, forces the receiver to resynchronize to the incoming data pattern, and starts a new test.

Example:

>RESULT:RESTART :restart a test

RESULT

Result Value **query**

RESULT? (result_name) :return the current value of the specified result

The **RESULT** query returns the current value of the specified (result_name). If the result is not applicable in the current test mode, no response is generated. If the result is applicable but "not ready," a special value is returned, as follows:

Result:	Not-Ready Value:
BIT_ERRS	-1
AVG_BER	-1.0E+0
BER	-1.0E+0
BLOCKS	-1
BLK_ERRS	-1
AVG_BLER	-1.0E+0
CHAR_ERR	-1
PAT_SLIP	-1
GERR_SEC	-1
G_EFS	-1
G_%EFS	"NOT READY"
DEG_MIN	-1
%DEG_MIN	"NOT READY"
SES	-1
%SES	"NOT READY"
G_VFS	-1
G_%VFS	"NOT READY"

RESULT (Continued)

Result Value **query**

Result:	Not-Ready Value:
UNA_SEC	-1
AVL_SEC	-1
%AVL_SEC	"NOT READY"
PATL_SEC	-1
%PAT_SEC	-1.0E+0
SIGL_SEC	-1
%ACT_SEC	"NOT READY"
SVS	-1
%SVS	"NOT READY"
TIME	"NOT READY"
DATE	"NOT READY"
ELAP_SEC	-1
EA_SEC	-1
ERR_EAS	-1
EF_EAS	-1
DELAY	"NOT READY"
MAX_JTR	"NOT READY"
1SEC_JTR	"NOT READY"
JTR_HITS	-1
+WNDR	"NOT READY"
-WNDR	"NOT READY"
PP_WNDR	"NOT READY"
15m_WNDR	"NOT READY"
24h_WNDR	"NOT READY"
TIE_WNDR	"NOT READY"
RCV_FREQ	"NOT READY"
GEN_FREQ	"NOT READY"

RESULT (Continued)

Result Value

query

Result:	Not-Ready Value:
%_MASK	"NOT READY"
S/A_nnn	"NOT READY"
RCV_BYTE	"NOT READY"
RCV_CODE	"NOT READY"
PWR_STAT	"NOT READY"
R_LVL_V	-1.00
R_LVL_dB	-1.00
TX_INFO	"NOT READY"
RCV_INFO	"NOT READY"
TEI	"NOT READY"
VCE_TEI	"NOT READY"
DATA_TEI	"NOT READY"
+_LVL_V	"NOT READY"
+_LVL_dB	"NOT READY"
-_LVL_dB	"NOT READY"
-_LVL_V	"NOT READY"
LVL_dBm	"NOT READY"
PP_LVL_V	"NOT READY"
DAT_RATE	-1.00
SMPX_CUR	"NOT READY"
BPVs	-1
AVG_BPVR	-1.0E+0
BPV_RATE	-1.0E+0
CODE_ERR	-1
AVG_CER	-1.0E+0
CER	-1.0E+0

RESULT (Continued)

Result Value **query**

Result:	Not-Ready Value:
VF_EAS	-1
%VF_EAS	"NOT READY"
FRA_ERR	-1
AVG_FER	-1.0E+0
FE_RATE	-1.0E+0
FRA_LOSS	-1
CRC_ERR	-1
AVG_CRC	-1.0E+0
CRC_E_RT	-1.0E+0
FAS_ERR	-1
AVG_FAS	-1.0E+0
FAS_E_RT	-1.0E+0
1SEC_CRC	-1
L2_F_ERR	-1
FCS_ERR	-1
L2_ABORT	-1
L2_FRMS	-1
REJ_FRM	-1
FRMR	-1
MAX_ZERO	-1
BIT_SLIP	-1
RCV_BOM	"NOT READY"
RX_ABCD	"NOT READY"
PAT_LOSS	-1
DAT_LOSS	-1
CLK_LOSS	-1

RESULT (Continued)

Result Value

query

Result:	Not-Ready Value:
C-D_CHA	-1
PWR_LOSS	-1
SIG_LOSS	-1
YEL_SEC	-1
XSO_SEC	-1
AIS_SEC	-1
1S_DEN_S	-1
V_FR_SEC	-1

Example:

```
>RESULT? BIT_ERRS          :print value of bit errors
-1
```

STATUS

The STATUS commands and queries are used to set and read the status reporting registers, queues, and buffers. For additional information see Section 5, Status Registers.

The STATUS commands and queries are:

command	query
STATUS:CLEAR_ERR	STATUS:ERROR?
STATUS:CLEAR_PRINT	
	STATUS:INTF?
	STATUS:INTF_HIST?
STATUS:INTF_HIST_EN	STATUS:INTF_HIST_EN?
	STATUS:INTF_LOSS?
STATUS:INTF_LOSS_EN	STATUS:INTF_LOSS_EN?
	STATUS:LINE?
	STATUS:LINE_HIST?
STATUS:LINE_HIST_EN	STATUS:LINE_HIST_EN?
	STATUS:LINE_LOSS?
STATUS:LINE_LOSS_EN	STATUS:LINE_LOSS_EN?
	STATUS:PRINT?

CLEAR_ERR

Clear Error Queue **command**

STATUS:CLEAR_ERR :clear the error message queue

The **STATUS:CLEAR_ERR** command clears the error message queue.

Example:

```
>STATUS:ERROR?                    :read the latest error message
"ERROR(107): Unrecognized Command"
>STATUS:CLEAR_ERR                 :clear the error message
>STATUS:ERROR?                    :return the most recent error
                                   message
"No Errors"
```

CLEAR_PRINT

Clear Print Buffer **command**

STATUS:CLEAR_PRINT :clear remote control print buffer

The **STATUS:CLEAR_PRINT** command clears the remote control print buffer.

Example:

```
>STATUS:CLEAR_PRINT          :clear the RC print queue
>STATUS:PRINT?               :return the next printout in the
                              remote control buffer
"ERROR(219): Print buffer is empty."
```

ERROR

Error Buffer — query

STATUS:ERROR? :return the most recent error message from the error buffer

The **STATUS:ERROR?** query returns the most recent error message from the error buffer. A complete list of errors messages is available in Section 6. If no errors are present in the error buffer, the message **No Error** is returned. The response is an error message enclosed in double quotes.

Example:

>ASDF :ASDF is an unrecognized command

"ERROR(107): Unrecognized command."

>STATUS:ERROR? :return the most recent error message

"ERROR(107): Unrecognized command."

INTF

Interface Status Register — query

STATUS:INTF? :report the current value of the Interface Status Register

The **STATUS:INTF?** query reports the current value of the Interface Status Register. This register provides the status of various DTE/DCE signaling leads. A one (1) in any bit position indicates that the corresponding signaling lead is active. When read, the value returned is the total weight of the bits that are set high.

Table 4-6 describes the standard bit settings for this status register, along with their corresponding weight values. This status register is also described in Section 5.

NOTE: The bit settings shown in Table 4-6 do not apply to the DDS Local Loop Interface (Model 41131/41231). Please see Section 3 of the *DDS Local Loop Interface Manual* for a description of the bit settings that are valid for this interface.

Table 4-6
Interface Status Register

Bit	Weight	Name	Meaning
7	128	—	Not used. Always set to zero.
6	64	—	Not used. Always set to zero.
5	32	DSR/DTR	Set to one when the DSR (DCE emulation) or DTR (DTE emulation) signaling switch is set to ON.
4	16	RLSD/RTS	Set to one when the RLSD (DCE emulation) or RTS (DTE emulation) signaling switch is set to ON.

INTF (Continued)

Interface Status Register — query

Table 4-6
Interface Status Register (Continued)

Bit	Weight	Name	Meaning
3	8	RL/TM	Set to one when the RL (DCE emulation) or TM (DTE emulation) is set high.
2	4	LL/CTS	Set to one when the LL (DCE emulation) or CTS (DTE emulation) is set high.
1	2	DTR/DSR	Set to one when the DTR (DCE emulation) or DSR (DTE emulation) is set high.
0	1	RTS/RLSD	Set to one when the RTS (DCE emulation) or RLSD (DTE emulation) is set high.

Example:

```
>STATUS:INTF?
2
```

:read Interface Status Register
:assuming the FIREBERD is emulating a DCE, the value of 2 indicates that the DTR signaling lead is set high.

INTF_HIST

Interface History Status Register **query**

STATUS:INTF_HIST? :report the current value of the Interface History Status Register

The **STATUS:INTF_HIST?** query reports the value of the Interface History Status Register. This register provides a history of DTE/DCE signaling leads that have switched from off to on since this register was last read. When read, the value returned is the total weight of the bits that are set high (i.e., 1) at the time the byte is read. Reading the register resets all bits to zero.

Table 4-7 contains information on each bit and bit weight in the Interface History Status Register. Note that this status register is also described in Section 5.

Table 4-7
Interface History Status Register

Bit	Weight	Name	Meaning
7	128	—	Not used. Always set to zero.
6	64	—	Not used. Always set to zero.
5	32	DSR/DTR	Set to one when the DSR (DCE emulation) or DTR (DTE emulation) signaling switch is set to ON.
4	16	RLSD/RTS	Set to one when the RLSD (DCE emulation) or RTS (DTE emulation) signaling switch is set to ON.

INTF_HIST (Continued)

Interface History Status Register **query**

Table 4-7
Interface History Status Register (Continued)

Bit	Weight	Name	Meaning
3	8	RL/TM	Set to one when the RL (DCE emulation) or TM (DTE emulation) is set high.
2	4	LL/CTS	Set to one when the LL (DCE emulation) or CTS (DTE emulation) is set high.
1	2	DTR/DSR	Set to one when the DTR (DCE emulation) or DSR (DTE emulation) is set high.
0	1	RTS/RLSD	Set to one when the RTS (DCE emulation) or RLSD (DTE emulation) is set high.

Example:

```
>STATUS:INTF_HIST?          :read Interface History Status Register
2                           :assuming the FIREBERD is emulating a
                             DCE, the value of 2 indicates that the DTR
                             signaling lead is set high.
```

INTF_HIST_EN

Set Enable Mask **command/query**

STATUS:INTF_HIST_EN (mask) :set the Interface History Enable Status Register

STATUS:INTF_HIST_EN? :return the value of the Interface History Enable Status Register

The **STATUS:INTF_HIST_EN** command sets the enable mask for the Interface History Status Register bits. The Interface History Enable Register determines how the Interface Summary bit of the Status Byte Register is set. The Interface Summary bit is set when corresponding bits in the Interface History Register and the Interface History Enable Register are set.

The Interface History Enable Status Register contains a mask value for the bits to be enabled in the Interface History Register. Table 4-8 describes the Interface History Enable Status Register. Note that this status register is also described in Section 5.

Table 4-8
Interface History Enable Status Register

Bit	Weight	Name	Meaning
7	128	—	Not used. Always set to zero.
6	64	—	Not used. Always set to zero.
5	32	DSR/DTR	Set to one when the DSR (DCE emulation) or DTR (DTE emulation) signaling switch is set to ON.
4	16	RLSD/RTS	Set to one when the RLSD (DCE emulation) or RTS (DTE emulation) signaling switch is set to ON.

INTF_HIST_EN (Continued)

Set Enable Mask **command/query**

Table 4-8
Interface History Enable Status Register (Continued)

Bit	Weight	Name	Meaning
3	8	RL/TM	Set to one when the RL (DCE emulation) or TM (DTE emulation) is set high.
2	4	LL/CTS	Set to one when the LL (DCE emulation) or CTS (DTE emulation) is set high.
1	2	DTR/DSR	Set to one when the DTR (DCE emulation) or DSR (DTE emulation) is set high.
0	1	RTS/RLSD	Set to one when the RTS (DCE emulation) or RLSD (DTE emulation) is set high.

Example:

>STATUS:INTF_HIST_EN 7 :assuming the FIREBERD is emulating a DTE, the value of 7 enables the CTS, DSR, and RLSD signaling leads to set the Interface Summary bit (4 + 2 + 1)

>STATUS:INTF_HIST_EN? :read the Interface History Enable Register

7 :the Interface Summary bit is set if the CTS, DSR, or RLSD bits are set in the Interface History Register

INTF_LOSS

Read Interface Loss Register **query**

STATUS:INTF_LOSS? :report the current value of the Interface Loss Status Register

The **STATUS:INTF_LOSS?** query reports the current value of the Interface Loss Status Register. This register provides a history of signaling leads that have switched from on to off. A bit is set if this event has occurred since this register was last read. Table 4-9 describes the Interface Lost Status Register. Note that this status register is also described in Section 5.

Table 4-9
Interface Loss Status Register

Bit	Weight	Name	Meaning
7	128	—	Not used. Always set to zero.
6	64	—	Not used. Always set to zero.
5	32	DSR/DTR	Set to one when the DSR (DCE emulation) or DTR (DTE emulation) signaling switch is set to ON.
4	16	RLSD/RTS	Set to one when the RLSD (DCE emulation) or RTS (DTE emulation) signaling switch is set to ON.

INTF_LOSS (Continued)

Read Interface Loss Register **query**

Table 4-9
Interface Loss Status Register (Continued)

Bit	Weight	Name	Meaning
3	8	RL/TM	Set to one when the RL (DCE emulation) or TM (DTE emulation) is set high.
2	4	LL/CTS	Set to one when the LL (DCE emulation) or CTS (DTE emulation) is set high.
1	2	DTR/DSR	Set to one when the DTR (DCE emulation) or DSR (DTE emulation) is set high.
0	1	RTS/RLSD	Set to one when the RTS (DCE emulation) or RLSD (DTE emulation) is set high.

Example:

>STATUS:INTF_LOSS?
4

:read Interface Loss Status Register
:assuming the FIREBERD is emulating a DTE, the value of 4 indicates that the CTS signaling lead was switched from on to off since this register was last read.

INTF_LOSS_EN

Set Enable Mask **command/query**

STATUS:INTF_LOSS_EN (mask) :set the Interface Loss Enable Status Register

STATUS:INTF_LOSS_EN? :return the value of the Interface Loss Enable Status Register

The **STATUS:INTF_LOSS_EN** command sets the enable mask for the Interface Loss Status Register bits. The Interface Loss Enable Register determines how the Interface Summary bit of the Status Byte Register is set. The Interface Summary bit is set when corresponding bits in the Interface History Register and the Interface History Enable Register are set.

The Interface Loss Enable Status Register contains a mask value for the bits to be enabled in the Interface Loss Register. Table 4-10 describes the Interface Loss Enable Status Register. Note that this status register is also described in Section 5.

Table 4-10
Interface Loss Enable Status Register

Bit	Weight	Name	Meaning
7	128	—	Not used. Always set to zero.
6	64	—	Not used. Always set to zero.
5	32	DSR/DTR	Set to one when the DSR (DCE emulation) or DTR (DTE emulation) signaling switch is set to ON.
4	16	RLSD/RTS	Set to one when the RLSD (DCE emulation) or RTS (DTE emulation) signaling switch is set to ON.

INTF_LOSS_EN (Continued)

Set Enable Mask **command/query**

Table 4-10
Interface Loss Enable Status Register (Continued)

Bit	Weight	Name	Meaning
3	8	RL/TM	Set to one when the RL (DCE emulation) or TM (DTE emulation) is set high.
2	4	LL/CTS	Set to one when the LL (DCE emulation) or CTS (DTE emulation) is set high.
1	2	DTR/DSR	Set to one when the DTR (DCE emulation) or DSR (DTE emulation) is set high.
0	1	RTS/RLSD	Set to one when the RTS (DCE emulation) or RLSD (DTE emulation) is set high.

Example:

>STATUS:INTF_LOSS_EN 7 :assuming the FIREBERD is emulating a DTE, the value of 7 enables the CTS, DSR, and RLSD signaling leads to set the Interface Summary bit (4 + 2 + 1).

>STATUS:INTF_LOSS_EN? :read the Interface Loss Enable Register

7 :the Interface Summary bit is set if the CTS, DSR, or RLSD bits are set in the Interface Loss Register

LINE

Line Status Register — query

STATUS:LINE? :report the current value of the Line Status Register

The **STATUS:LINE?** query reports the current value of the Line Status Register. This register provides the current state of various signal indicators. When read, the value returned is the total weight of the bits that are high (i.e., set to 1) at the time the byte is read.

Table 4-11 describes each bit in the Line Status Register and its corresponding weight value. Note that this status register is also described in Section 5.

Table 4-11
Line Status Register

Bit	Weight	Name	Meaning
7	128	-	Not used. Always set to zero.
6	64	TEST COMPLETE	Set to one when the test is complete.
5	32	RCV CLK INV	Set to one when an inverted clock is received.
4	16	RCV DATA INV	Set to one when inverted data is received.
3	8	GEN CLK PRES	Set to one when generated clock is present.
2	4	PATTERN SYNC	Set to one when pattern sync is achieved.
1	2	FRAME SYNC	Set to one when frame sync is achieved.
0	1	SIGNAL PRESENT	Set to one when signal activity is detected at the input.

LINE (Continued)

Line Status Register — query

Example:

```
>STATUS:LINE?          :read line status register
37                     :the value of 37 indicates that the FIREBERD
                        6000 is receiving an inverted clock, is in
                        pattern sync, and detects signal presence
                        (37 = 32 + 4 + 1)
```

LINE_HIST

Line History Status Register — query

STATUS:LINE_HIST? :report the value of the Line History Status Register

The **STATUS:LINE_HIST?** query reports the value of the Line History Status Register. This register provides a history of signal indicators that have switched from off to on since this register was last read. When read, the value returned is the total weight of the bits that are high (i.e., set to 1). Reading the register resets all bits to zero.

Table 4-12 contains information on each bit and bit weight in the Line History Status Register. Note that this status register is also described in Section 5.

Table 4-12
Line History Status Register

Bit	Weight	Name	Meaning
7	128	-	Not used. Always set to zero.
6	64	TEST COMPLETE	Set to one when the test is complete.
5	32	RCV CLK INV	Set to one when an inverted clock is received.
4	16	RCV DATA INV	Set to one when inverted data is received.
3	8	GEN CLK PRES	Set to one when generated clock is present.
2	4	PATTERN SYNC	Set to one when pattern sync is achieved.
1	2	FRAME SYNC	Set to one when frame sync is achieved.
0	1	SIGNAL PRESENT	Set to one when signal activity is detected at the input.

LINE_HIST (Continued)

Line History Status Register — query

Example:

```
>STATUS:LINE_HIST?
28
```

:read the line history status register
:the value 28 indicates that the FIREBERD received inverted data, detected generator clock presence, and achieved pattern sync since this register was last read (28 = 16 + 8 + 4)

LINE_HIST_EN

Set Enable Mask **command/query**

STATUS:
LINE_HIST_EN (mask) :set the Line History Enable Status Register

STATUS:
LINE_HIST_EN? :return the value of the Line History Enable Status Register

The **STATUS:LINE_HIST_EN** command sets the enable mask for the Line History Status Register bits. The Line History Enable Register determines how the Line Summary bit of the Status Byte Register is set. The Line Summary bit is set when corresponding bits in the Line History Register and the Line History Enable Register are set.

The Line History Enable Status Register contains a mask value for the bits to be enabled in the Interface History Register. Table 4-13 describes the Line History Enable Status Register. Note that this status register is also described in Section 5.

Table 4-13
Line History Enable Status Register

Bit	Weight	Name	Meaning
7	128	-	Not used. Always set to zero.
6	64	TST COMP	Set to one when the test is complete.
5	32	RCV CLK INV	Set to one when an inverted clock is received.
4	16	RCV DATA INV	Set to one when inverted data is received.
3	8	GEN CLK PRES	Set to one when generated clock is present.
2	4	PATTERN SYNC	Set to one when pattern sync is achieved.
1	2	FRAME SYNC	Set to one when frame sync is achieved.
0	1	SIGNAL PRESENT	Set to one when signal activity is detected at the input.

LINE_HIST_EN (Continued)

Set Line History Enable Register **command/query**

Example:

```
>STATUS:LINE_HIST_EN 37      :the value of 37 enables inverted clock,
                              pattern sync, and signal presence to
                              set the Line Summary bit (32 + 4 + 1)

>STATUS:LINE_HIST_EN?        :read Line History Enable Register
37                            :the Line Summary bit is set if the
                              inverted clock, pattern sync, or signal
                              presence bit is set in the Line History
                              Register
```

LINE_LOSS

Read Line Loss Register **query**

STATUS:LINE_LOSS? :report the current value of the Line Loss Status Register

The **STATUS:LINE_LOSS?** query reports the current value of the Line Loss Status Register. This register provides a history of signal loss, synchronization loss, and generated clock loss. A bit is set if any of these events have occurred since the register was last read. Table 4-14 describes the Line Loss Status Register. Note that this status register is also described in Section 5.

Table 4-14
Line Loss Status Register

Bit	Weight	Name	Meaning
7	128	—	Not used. Always set to zero.
6	64	—	Not used. Always set to zero.
5	32	—	Not used. Always set to zero.
4	16	—	Not used. Always set to zero.
3	8	GEN CLK LOSS	Generated clock was lost since this register was last read.
2	4	PATTERN SYNC LOSS	Pattern synchronization was lost since this register was last read.
1	2	FRAME SYNC LOSS	Frame synchronization was lost since this register was last read.
0	1	SIGNAL LOSS	Signal presence was lost since this register was last read.

LINE_LOSS (Continued)

Read Line Loss Register **query**

Example:

```
>STATUS:LINE_LOSS?          :read the line loss status register
14                          :the value 14 indicates that the
                             FIREBERD lost the generator clock,
                             pattern synchronization, and frame
                             synchronization since this register was
                             last read (14 = 8 + 4 + 2)
```

LINE_LOSS_EN

Set Enable Mask **command/query**

STATUS:LINE_LOSS_EN (mask) :set the Line Loss Enable Status Register

STATUS:LINE_LOSS_EN? :return the value of the Line Loss Enable Status Register

The **STATUS:LINE_LOSS_EN** command sets the enable mask for the Line Loss Status Register bits. The Line Loss Enable Register determines how the Line Summary bit of the Status Byte Register is set. The Line Summary bit is set when corresponding bits in the Line History Register and the Line History Enable Register are set.

The Line Loss Enable Status Register contains a mask value for the bits to be enabled in the Interface History Register. Table 4-15 describes the Line Loss Enable Status Register. Note that this status register is also described in Section 5.

Table 4-15
Line Loss Enable Status Register

Bit	Weight	Name	Meaning
7	128	—	Not used. Always set to zero.
6	64	—	Not used. Always set to zero.
5	32	—	Not used. Always set to zero.
4	16	—	Not used. Always set to zero.
3	8	GEN CLK LOSS	Generated clock was lost since this register was last read.
2	4	PATTERN SYNC LOSS	Pattern synchronization was lost since this register was last read.
1	2	FRAME SYNC LOSS	Frame synchronization was lost since this register was last read.
0	1	SIGNAL LOSS	Signal presence was lost since this register was last read.

LINE_LOSS_EN (Continued)

Set Enable Mask **command/query**

Example:

```
>STATUS:LINE_LOSS_EN 37      :the value of 37 enables inverted clock,
                              pattern sync, and signal presence to
                              set the Line Summary bit (32 + 4 + 1)
>STATUS:LINE_LOSS_EN?        :read Line History Enable Register
37                            :the Line Summary bit is set if the
                              inverted clock, pattern sync, or signal
                              presence bit is set in the Line Loss
                              Register
```

PRINT

Remote Control Print Buffer **query**

STATUS:PRINT? :return the next printout in the remote control buffer

The **STATUS:PRINT?** query responds with the next printout in the remote control print buffer.

Example:

```
>PRINT:DEVICE REMOTE          :enable printing to remote
                               control print buffer
>PRINT:CONTROLS               :print controls printout
>*STB?                        :report that the Print Available
                               bit is set
2
>STATUS:PRINT?                :obtain printout from remote
                               control print buffer
.
.
.                             :a controls print is printed at
                               your terminal
```

SECTION 5
STATUS REPORTING

5.1 INTRODUCTION

The FIREBERD 6000 uses a set of internal registers, queues, and buffers to store information about test status, line conditions, alarms, error conditions, and printer data. This information can be accessed using some of the FIREBERD's queries.

5.2 REGISTER GROUPS

The FIREBERD 6000 has four groups of registers: Status Byte Registers, Event Status Registers, Line Status Registers, and Interface Status Registers (Figure 5-1).

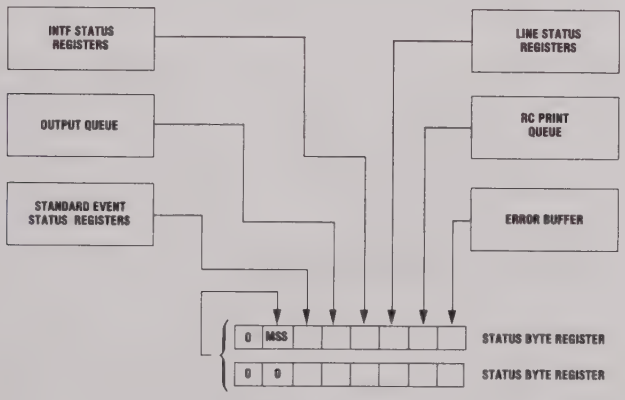

Figure 5-1
FIREBERD 6000 Registers

Each group of registers captures specific events. Within each group of registers is a status register which is composed of 8 bits representing up to 8 events. When a specific event occurs, the corresponding bit is set to 1. In this way, the bits in each status register can provide the user with information about the test in progress.

Each group of registers reports both **present and past** events. Status registers convey present signal status; history and loss registers report changes that have occurred since these registers were last read. A history bit is set when a signal is switched from OFF to ON; a loss bit is set when a signal is switched from ON to OFF.

Additionally, each group of registers contains one or two writable eight-bit enable registers. The bits in each enable register can be programmed to report selected events in the corresponding status register. Setting the bits in the enable register controls the generation of the summary bit for each register group.

The summary bit is set when a status register bit and its corresponding register bit are both set. In the Interface and Line Register groups, the summary bit is set when:

- A loss bit and its corresponding loss enable bit are set
- A history bit and its corresponding history enable bit are set

Registers may be read using an associated query. Enable registers may be written to using an associated command. The *CLS command clears all event registers, buffers, and queues except the output queue.

5.2.1 An Example of Registers Operation

In Figure 5-2, the value 36 has been written to Enable Register 1. This value sets bits 2 and 5 of Enable Register 1. Since bit 5 of both Status Register 1 and Enable Register 1 is set, the summary bit is set in Status Register 2.

This process is duplicated with each group of registers until all of the summary information is directed to the Status Byte register. The information contained in the Status Byte register enables the SRQ signal on the IEEE-488 bus. In this way, programmers can have the FIREBERD signal the occurrence of an event of interest.

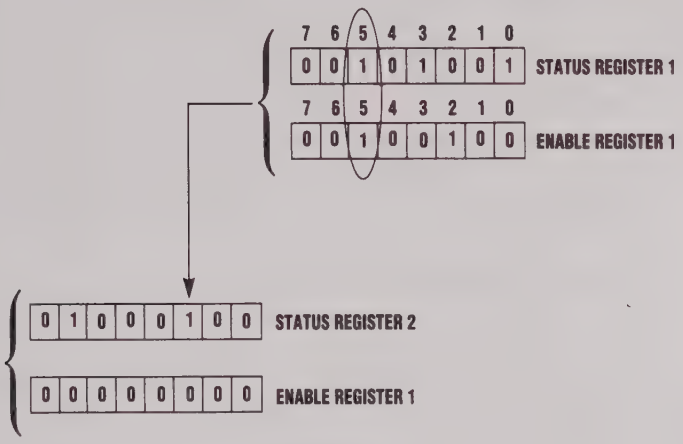
7 6 5 4 3 2 1 0
0 0 1 0 1 0 0 1
STATUS REGISTER 1
7 6 5 4 3 2 1 0
0 0 1 0 0 1 0 0
ENABLE REGISTER 1
0 1 0 0 0 1 0 0
STATUS REGISTER 2
0 0 0 0 0 0 0 0
ENABLE REGISTER 1

Figure 5-2
Register Operation

5.3 QUEUE and BUFFER OPERATION

5.3.1 Output Queue

All responses generated by the FIREBERD 6000 are placed in the Output Queue. In RS-232 remote control operation, the contents of the Output Queue are immediately transmitted to the terminal. In IEEE-488 remote control operation, responses are stored in the Output Queue until the FIREBERD is addressed to TALK. When a response is in the Output Queue, the Message Available indicator (bit 4 of the Status Byte Register) is set to 1.

5.3.2 Print Queue

Printouts generated by the FIREBERD 6000 can be sent to an internal buffer called the RC Print Queue. A printout in the RC Print Queue sets the Print Available indicator (bit 1 of the Status Byte Register) to 1. This activity is enabled by setting the printer driver to REMOTE. Printouts are read using the STATUS:PRINT? query.

5.3.3 Error Buffer

The Error Buffer is an internal buffer that stores the last generated error message. It is read using the STATUS:ERROR? query and is cleared using the STATUS:CLEAR_ERR command. When an error message is present in the Error Buffer, the Error Available indicator (bit 0 of the Status Byte Register) is set to 1. Errors messages are also transmitted directly to the terminal in RS-232 terminal mode operation.

5.4 STATUS BYTE REGISTER GROUP

The Status Byte Register group consists of two eight-bit registers: Status Byte Register and Status Byte Enable Register. This register group summarizes the data structures reporting to it (see Figure 5-3).

5.4.1 Status Byte Register

The Status Byte Register contains summary bits from the data structures reporting to it. Each bit reports whether a specific event or condition has occurred. The FIREBERD 6000 defines seven of the eight bits in the Status Byte Register. The other bit is always set to zero.

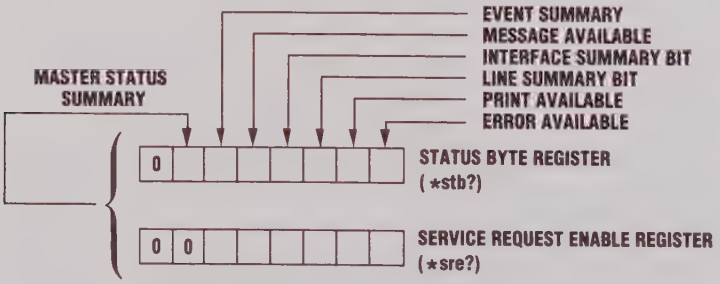

Figure 5-3
Status Byte Register Group

The controller can read the Status Byte Register using a serial poll or using the *STB query. When a serial poll is used, bit 6 is returned as a Request Service (RQS); when the *STB? query is used, bit 6 is returned as a Master Summary Status (MSS).

When the Status Byte Register is read, a weighted integer is returned. This integer represents the decimal equivalent of the enabled bits in the register. To determine the status of the bits in the Status Byte Register, convert the returned value to binary.

Bit 7 Not used. Always zero.

Bit 6 **MSS - Master Status Summary**
Represents the summary of all other bits in the Status Byte Register. The corresponding enabling bit in the Service Request Enabling Register (*SRE) is not programmable. When this bit is set to 1, an SRQ is asserted on the IEEE-488 bus.

Bit 5 **ESB - Event Status Bit**
Indicates that an enabled event has occurred in the Standard Event Status Register. Read the Standard Event Status Register to determine the specifics of the event.

Bit 4 **MAV - Message Available**
Indicates that a response is available in the output queue. The FIREBERD does not talk on an IEEE-488 bus until the MAV bit is set to 1.

Bit 3 **INT - Interface Summary Bit**
Indicates a status change in the Interface Status Register. Read the Interface History and Loss Registers to determine the specifics of the status change.

Bit 2 **LSB - Line Summary Bit**
Indicates a status change in the Line Status Register. Read the History and Loss Registers to determine the specifics of the status change.

Bit 1 **PAB - Print Buffer Available**
Indicates that a printout is available in the remote control print buffer (see STATUS:PRINT?).

Bit 0 **EAB - Error Buffer Available**
Indicates that an error message is available in the error buffer (see STATUS:ERROR? and STATUS:CLEAR_ERR).

5.4.2 Status Byte Enable Register

Status Byte Enable is an 8-bit register that determines which bits of the Status Byte Register can cause the MSS bit to be set. The MSS bit is used to assert SRQ when using an IEEE-488 bus. This means that the Service Request Enable register effectively determines when the FIREBERD 6000 will request service. Use the *SRE command to write to the Service Request Enable Register.

5.5 EVENT STATUS REGISTER GROUP

The Event Status Register group detects specific IEEE-488-defined events (Figure 5-4). It consists of two eight-bit registers: the Event Status Register and the Event Status Enable Register.

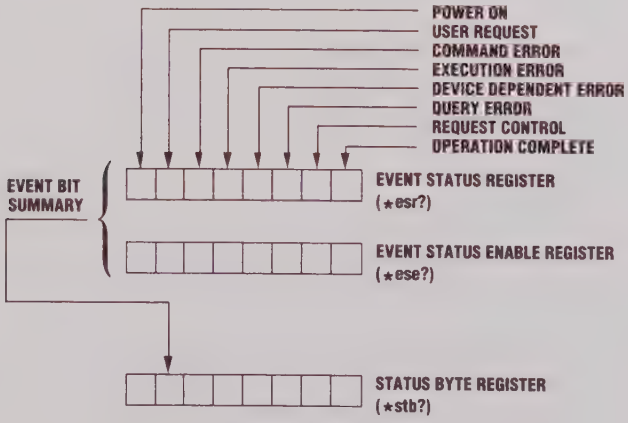

Figure 5-4
Event Status Register Group

5.5.1 The Event Status Register

The Event Status Register (ESR) assigns specific IEEE-488.2-defined events to each of the bits within the register. A bit is set in the Event Status Register when the event that corresponds to that bit has occurred since the register was last read or cleared. This register is cleared each time it is read.

Bit 7 **PON - Power On**
Power has been cycled OFF and ON since the last time the Event Status Register was read.

Bit 6 **URQ - User Request**
Not used. Always reads zero.

Bit 5 **CME - Command Error**
Indicates that a command received by the FIREBERD 6000 was not executed because it was not understood.

Bit 4 **EXE - Execution Error**
Indicates that the FIREBERD is unable to execute a command.

Bit 3 **DDE - Device-Dependent Error**
Not used. Always reads zero.

Bit 2 **QYE - Query Error**
Indicates that the controller has attempted to read the FIREBERD's output queue when nothing was available to be read, or that data in that queue has been lost.

Bit 1 **RQC - Request Control**
Not used. Always reads zero.

Bit 0 **OPC - Operation Complete**
Indicates that all pending operations are complete (see the *OPC command in Section 4).

5.5.2 The Event Status Enable Register

Event Status Enable is an eight-bit register that allows the controller to enable or disable each bit in the Event Status Register. When a bit in the Event

Status Enable Register is set to 1, the corresponding bit in the Event Status Register is enabled. When an enabled bit in the Event Status Register is set to 1, the ESB bit in the Status Byte Register is also set to 1.

5.6 LINE SUMMARY REGISTER GROUP

The Line Summary register group consists of five registers: the Line Status Register, Line History Register, Line History Enable Register, Line Loss Register, and Line Loss Enable Register (Figure 5-5).

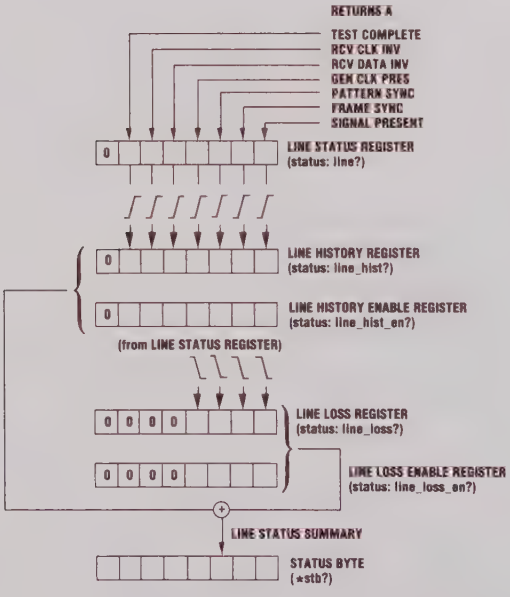

Figure 5-5
Line Summary Register Group

5.6.1 The Line Status Register

The Line Status Register gives the current state of various signal indicators. The Line Status Register is read using the query: STATUS:LINE?.

Bit 7 Not used. Always zero.

Bit 6 **TEST COMPLETE**
Set to one when the test is complete.

Bit 5 **RCV CLK INV**
Set to one when an inverted clock is received.

Bit 4 **RCV DATA INV**
Set to one when inverted data is received.

Bit 3 **GEN CLK PRES**
Set to one when generated clock is present.

Bit 2 **PATTERN SYNC**
Set to one when pattern sync is achieved.

Bit 1 **FRAME SYNC**
Set to one when the FIREBERD detects frame synchronization.

Bit 0 **SIGNAL PRESENT**
Set to one when an input signal is present.

5.6.2 The Line History Register

The Line History Register provides the history of various signal indicators since this register was last read. When a bit is set, it generally indicates that an error has occurred or synchronization was lost during the interval since the register was last read. The Line History Register is read using the query: STATUS:LINE HIST?.

Bit 7 Not used. Always zero.

Bit 6 **TEST COMPLETE**
Set to one when the test is complete.

Bit 5 **RCV CLK INV**
Set to one when an inverted clock is received.

Bit 4 **RCV DATA INV**
Set to one when inverted data is received.

Bit 3 **GEN CLK PRES**
Indicates that the generated clock was present since this register was last read.

Bit 2 **PATTERN SYNC**
Indicates that the FIREBERD synchronized to the received pattern since this register was last read.

Bit 1 **FRAME SYNC**
Indicates that frame synchronization was detected since this register was last read.

Bit 0 **SIGNAL PRESENT**
Indicates that signal presence was detected since this register was last read.

5.6.3 The Line History Enable Register

The Line History Enable Register determines how the Line Summary bit of the Status Byte Register is set. The Line Summary bit is set when corresponding bits in the Line History Register and the Line History Enable Register are set. The Line History Enable Register is read with the query: STATUS:LINE_HIST_EN?.

5.6.4 The Line Loss Register

The Line Loss Register provides a history of signal loss, synchronization loss, and generated clock loss. A bit is set if any of these events have occurred since the register was last read. The Line Loss Register is read using the query STATUS:LINE_LOSS?.

Bit 7 Not used. Always set to zero.

Bit 6 Not used. Always set to zero.

Bit 5 Not used. Always set to zero.

Bit 4 Not used. Always set to zero.

Bit 3 **GEN CLK LOSS**
Indicates that the generated clock was lost since this register was last read.

Bit 2 **PATTERN SYNC LOSS**
Indicates that the FIREBERD lost synchronization to the received pattern since this register was last read.

Bit 1 **FRAME SYNC LOSS**
Indicates that frame synchronization was lost since this register was last read.

Bit 0 **SIGNAL LOSS**
Indicates that signal presence was lost since this register was last read.

5.6.5 The Line Loss Enable Register

The Line Loss Enable Register determines how the Line Summary bit of the Status Byte Register is set. The Line Summary bit is set when corresponding bits in the Line Loss Register and the Line Loss Enable Register are set. Thus, either the Line History Register or Line Loss Register determine how the Line Summary bit is set. The Line Loss Enable Register is read using the query STATUS:LINE_LOSS_EN?.

5.7 INTERFACE STATUS REGISTER GROUP

The Interface Status register group consists of five registers: the Interface Status Register, Interface History Register, Interface History Enable Register, Interface Loss Register, and Interface Loss Enable Register (Figure 5-6).

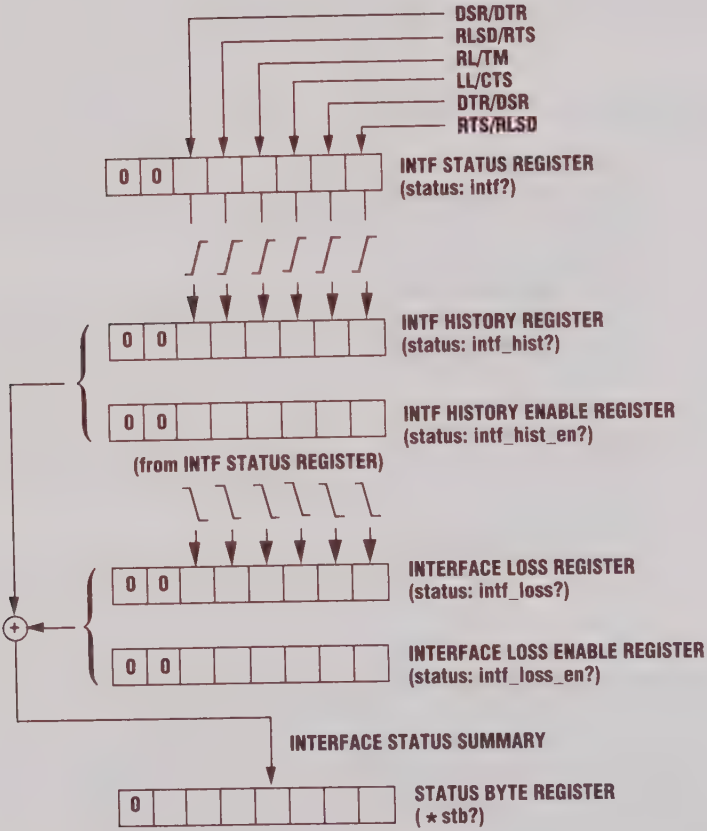

Figure 5-6
Interface Status Register

5.7.1 The Interface Status Register

The Interface Status Register indicates the current status of the interface signal. A one (1) in any bit position indicates that the corresponding signal is active. The Interface Status Register is read using the query: STATUS:INTF?.

NOTE: The bit settings immediately following do not apply to the DDS Local Loop Interface (Model 41131/41231). Please see Section 3 of the *DDS Local Loop Interface Manual* for a description of the bit settings that are valid for this interface.

Bit 7 Not used. Always zero.

Bit 6 Not used. Always zero.

Bit 5 **DSR/DTR**
Set to one when the DSR (DCE emulation) or DTR (DTE emulation) signaling switch is set to ON.

Bit 4 **RLSD/RTS**
Set to one when the RLSD (DCE emulation) or RTS (DTE emulation) signaling switch is set to ON.

Bit 3 **RL/TM**
Set to one when the RL (DCE emulation) or TM (DTE emulation) signaling lead is set high.

Bit 2 **LL/CTS**
Set to one when the LL (DCE emulation) or CTS (DTE emulation) signaling lead is set high.

Bit 1 **DTR/DSR**
Set to one when the DTR (DCE emulation) or DSR (DTE emulation) signaling lead is set high.

Bit 0 **RTS/RLSD**
Set to one when the RTS (DCE emulation) or RLSD (DTE emulation) signaling lead is set high.

5.7.2 The Interface History Register

The Interface History Register provides the history of the interface signal since this register was last read. A bit in this register is set when the corresponding bit in the Interface Status register changes from 0 (idle) to 1 (active). The bit in the Interface History Register remains set until this register is read. The Interface History Register is read using the query: STATUS:INTF HIST?.

Bit 7 Not used. Always zero.

Bit 6 Not used. Always zero.

Bit 5 **DSR/DTR**
Set to one when the DSR (DCE emulation) or DTR (DTE emulation) signaling switch is set to ON.

Bit 4 **RLSD/RTS**
Set to one when the RLSD (DCE emulation) or RTS (DTE emulation) signaling switch is set to ON.

Bit 3 **RL/TM**
Set to one when the RL (DCE emulation) or TM (DTE emulation) signaling lead is set high.

Bit 2 **LL/CTS**
Set to one when the LL (DCE emulation) or CTS (DTE emulation) signaling lead is set high.

Bit 1 **DTR/DSR**
Set to one when the DTR (DCE emulation) or DSR (DTE emulation) signaling lead is set high.

Bit 0 **RTS/RLSD**
Set to one when the RTS (DCE emulation) or RLSD (DTE emulation) signaling lead is set high.

5.7.3 The Interface History Enable Register

The Interface History Enable Register determines how the Interface Summary bit of the Status Byte Register is set. The Interface Summary bit is set when corresponding bits in the Interface History Register and the Interface History Enable Register are set. The Interface History Enable Register is read with the query: STATUS:INTF_HIST_EN?.

5.7.4 The Interface Loss Register

The Interface Loss Register provides a history of interface signal losses. A bit is set if any interface signal has been lost since the register was last read. The Interface Loss Register is read using the query STATUS:INTF_LOSS?.

Bit 7 Not used. Always zero.

Bit 6 Not used. Always zero.

Bit 5 **DSR/DTR**
Set to one when the DSR (DCE emulation) or DTR (DTE emulation) signaling switch is set to ON.

Bit 4 **RLSD/RTS**
Set to one when the RLSD (DCE emulation) or RTS (DTE emulation) signaling switch is set to ON.

Bit 3 **RL/TM**
Set to one when the RL (DCE emulation) or TM (DTE emulation) signaling lead is set high.

Bit 2 **LL/CTS**
Set to one when the LL (DCE emulation) or CTS (DTE emulation) signaling lead is set high.

Bit 1 **DTR/DSR**
Set to one when the DTR (DCE emulation) or DSR (DTE emulation) signaling lead is set high.

Bit 0 **RTS/RLSD**
Set to one when the RTS (DCE emulation) or RLSD (DTE emulation) signaling lead is set high.

5.7.5 The Interface Loss Enable Register

The Interface Loss Enable Register determines how the Interface Summary bit of the Status Byte Register is set. The Interface Summary bit is set when corresponding bits in the Interface Loss Register and the Interface Loss Enable Register are set. Thus, either the Interface History Register or Interface Loss Register determine how the Interface Summary bit is set.

SECTION 6
ERROR MESSAGES

6.1 INTRODUCTION

An error message indicates an error in an unprocessed remote control command. Error messages are generated according to the FIREBERD 6000 remote control mode of choice: RS-232 terminal, RS-232 computer, or IEEE-488.

RS-232 Terminal Mode

In terminal mode, the FIREBERD operates interactively. Therefore, error messages are transmitted to the terminal as soon as they occur. The error message is also placed in the error buffer, causing the Error Available indicator (bit 0 of the Status Byte Register) to be set.

RS-232 Computer Mode

In computer mode, the FIREBERD operates with a programmed computer which requires predictable results. Therefore, error messages are not sent to the computer when they occur, but are instead sent to the error buffer. This causes the Error Available indicator (bit 0 of the Status Byte Register) to be set.

IEEE-488 Computer Mode

When the FIREBERD 6000 operates via an IEEE-488 bus, error messages are sent to the error buffer. This causes the Error Available indicator (bit 0 of the Status Byte Register) to be set.

The error buffer holds one error message at a time. The error message currently held is the last error message generated. The error buffer is read using the STATUS:ERROR? query and is cleared using the STATUS:CLEAR_ERR command. When an error message is present in the error buffer, the Error Available indicator (bit 0 of the Status Byte Register) is set. In computer mode, error message must be cleared before issuing the next command.

6.2 ERROR MESSAGES

The following is a complete list of error messages reported by the FIREBERD 6000.

6.2.1 Command Errors

Argument Too Long (100)
Cause: An invalid command argument is specified.

Invalid Argument (101)
Cause: A command argument that does not match the expected argument syntax is specified.

Invalid Hexadecimal Argument (102)
Cause: A command argument that does not match the syntax of the expected hexadecimal argument is specified.

Invalid Numeric Argument (103)
Cause: The FIREBERD does not recognize the argument entered in the command line because it is not in numeric format.

String Argument Expected (104)
Cause: A command is issued without the argument enclosed in quotes.

String Too Long (105)
Cause: A specified command contains a string with more than 254 characters.

Syntax Error (106)
Cause: The FIREBERD does not recognize the input as a valid command. This error message may appear when a comma or semicolon is missing, or when a command is misspelled.

Unrecognized Command (107)
Cause: An unrecognizable command or query was received.

6.2.2 Execution Errors

An empty slot cannot be selected (200)
Cause: The specified interface slot is empty.

Asynchronous frequency contention (201)
Cause: The specified frequency is invalid in asynchronous timing mode.

Asynchronous pattern contention (202)
Cause: The specified pattern is invalid in asynchronous timing mode.

Cannot set while line is active (203)
Cause: A parameter change is not allowed while the line is active.

Device in use (204)
Cause: The printer driver specified is also the one used by remote control.

Illegal phone number (205)
Cause: The data specified does not constitute a legal phone number.

Illegal result name (206)
Cause: The data specified does not constitute a legal result.

Illegal type name (207)
Cause: The data specified does not constitute a legal type.

Illegal user message string (208)
Cause: The specified user message string is either not in the correct format or contains ASCII characters which are invalid.

Illegal user message byte position (209)
Cause: The byte position is not within the boundaries of the specified user message.

Illegal value (210)
Cause: The value specified with a command is not within the legal range of the requested function.

Inconsistent with current configuration (211)

Cause: The requested function is not applicable in the FIREBERD 6000's current mode.

Inconsistent with interface or interface mode (212)

Cause: The requested command is not supported by the current interface, or is not applicable or available in the current interface mode.

ISU-6000 is installed (213)

Cause: A command to select the external interface is specified while the ISU-6000 is installed.

ISU-6000 is not installed (214)

Cause: An ISU-6000 command is specified when the option is not installed.

ISU-2000 is not in remote mode (215)

Cause: A remote command was issued for the ISU-2000 while in local mode.

Jitter generation contention (216)

Cause: The command specified caused contention in the jitter generator.

Option not installed (217)

Cause: The specified command requires an option that is not installed.

Presynchronization/power loss unavailable (218)

Cause: Synchronization losses or power losses have not occurred.

Print buffer is empty (219)

Cause: A command to read the print buffer is issued when no printouts are available.

Remote control using RS232 (220)

Cause: A change in an RS-232 parameter is encountered while remote control is using the RS-232 driver.

Result not applicable (221)

Cause: The result specified is not applicable in the current timing or interface mode.

Software error (222)

Cause: An internal software error has occurred.

Synchronous pattern contention (223)

Cause: The pattern specified is invalid when synchronous timing is selected.

Unrecognized parameter (224)

Cause: The FIREBERD recognizes but cannot interpret the command parameter specified.

6.2.3 Query Errors

Response Interrupted (300)

Cause: The controller sent the FIREBERD a command or query before reading the entire response from a previous query. The response of the previous query is discarded.

Deadlock detected (301)

Cause: When a very long program message produces a very long response message, the commands in the input buffer. If this condition occurs, the entire response message is discarded. All commands sent are processed normally.

Command Unterminated (302)

Cause: The controller tried to read a response before a complete program message had been sent to the FIREBERD. No response is sent.

SECTION 7
IEEE-488.2 SPECIFICATION REQUIREMENTS

7.1 INTRODUCTION

This section addresses the Device Documentation Requirements listed in Section 4.9 of the IEEE-488.2 Specification (ANSI/IEEE Std 488.2-1987). The italicized items 1-23 correspond to the specific information required by the IEEE-488.2 Specification; the information supplied after each of these items responds to these requirements. Some of the responses reference sections of this manual where the information can be obtained.

NOTE: In this section, "device" refers to the FIREBERD 6000.

7.2 IMPLEMENTATION OF IEEE STANDARD 488.2-1987

All devices shall supply information to the user about how the device has implemented this standard. This information shall include:

1. A list of IEEE-488.1 Interface Function subsets implemented.

The 488.1 Interface Functions subsets are implemented in the FIREBERD 6000 as follows:

SH1:	Source Handshake capability
AH1:	Acceptor Handshake capability
T5:	Basic Talker capability with Serial Poll, Talk Only, and Unaddress if MLA
L4:	Basic Listener capability with Unaddress if MTA
SR1:	Status Request capability
RL1:	Remote/Local capability
DC1:	Device Clear capability
DT1:	Device Trigger capability
PP1:	Parallel Poll capability
CO:	No controller capability
E2:	Three-state drivers used

2. *A description of when a user-initiated address change is recognized by the device.*

   If the address is changed using the DIP switches on the interface, the change takes effect immediately. The valid address range is 0-30.

   NOTE: Verify that this command is executed before attempting to communicate with the FIREBERD at its new address.

3. *A description of the device settings at power-on and any commands which modify the power-on setting.*

   All of the device settings are preserved from power-off to power-on; thus, all commands which configure the FIREBERD 6000 will modify the power-on settings. The FIREBERD is shipped with the factory-default settings listed under the *RST command. These settings are restored when the user performs the CLEAR NOVRAM operation or when the *RST command is executed.

   NOTE: The RS-232 remote control mode always powers-on in the Computer mode.

4. *A description of message exchange options.*

   a. *The size and behavior of the input buffer.*

      The input buffer is 100 bytes long. This does not restrict the length of a program message to 100 bytes, since program messages are parsed as they are received.

   b. *Which queries return more than one <RESPONSE MESSAGE UNIT>.*

      No queries return more than one <RESPONSE MESSAGE UNIT>.

   c. *Which queries generate a response when parsed.*

      All queries generate a response when parsed.

d. Which queries generate a response when read.

No queries wait until the device is read to generate a response.

e. Which commands are coupled.

The configuration selection commands (CONFIG:TIMING:MODE, CONFIG:FREQ, and CONFIG:PATTERN) and the jitter generator commands (JITTER:GEN:MOD, JITTER:GEN:FREQ, and JITTER:GEN:AMP) are coupled. For more information on these commands refer to Section 4.

5. *A list of functional elements used in constructing the device-specific commands, including whether <compound command program header> elements are used.*

<Compound command program header> elements are used. Each functional element is listed in Section 4.

6. *A description of any buffer size limits related to block data.*

Block data elements are not used.

7. *A list of the <PROGRAM DATA> elements which may appear within an <expression>, as well as the maximum sub-expression nesting depth. Any additional syntax restrictions which the device may place on the <expression> shall also be included.*

<expression>s are not implemented.

8. *A description of the response syntax for every query.*

This information is offered in Section 4.

9. *A description of any device-to-device message transfer traffic which does not follow the rules for <RESPONSE MESSAGE> elements.*

No device-to-device message transfer traffic is implemented.

10. *The size of any block data responses.*

    No block data responses are implemented.

11. *A list of common commands and queries which are implemented.*

    See Section 4, COMMON COMMANDS.

12. *A description of the state of the device after successful completion of the Calibration query.*

    The Calibration query is not implemented.

13. *The maximum length of the block used to define the trigger macro, if *DDT is implemented.*

    *DDT is not implemented.

14. *The maximum length of macro labels, the maximum length of the block used to define a macro, and how recursion is handled during macro expansion, if the macro commands are implemented.*

    Macro commands are not implemented.

15. *A description of the response to the identification common query, *IDN?*

    See the command *IDN, described in Section 4.

16. *The size of the protected user data storage area, if the *PUD command and *PUD? query are implemented.*

    *PUD and *PUD? are not implemented.

17. *The size of the resource description, if the *RDT command and *RTD? query are implemented.*

    *RDT and *RTD? are not implemented.

18. *A description of the states affected by *RST, *LRN?, *RCL, and *SAV.*

    These commands all affect the same set of instrument settings. Refer to the instrument settings listed under the *RST command in Section 4.

19. *A description of the scope of the self-test performed by the *TST? query.*

    See the *TST command, described in Section 4.

20. *A description of additional status data structures used in the devices status reporting.*

    Section 5 (Status Reporting) is dedicated to this topic.

21. *For each command, a statement describing whether it is sequential or overlapped.*

    All commands are sequential. None overlap.

    NOTE: The execution order is modified in the case of coupled commands, but it is still sequential (i.e., not overlapped).

22. *For each command the device documentation shall specify the functional criteria that are met when an operation complete message is generated in response to that command.*

    When the **OPERATION COMPLETE** message is generated, the function of any given command has been performed. This means that the general description of each command, given in Section 4, constitutes the criteria met when this message is generated.

Telecommunications
Techniques
Corporation
A DYNATECH COMPANY

20410 Observation Drive
Germantown, MD 20876
Tel. (301) 353-1550
Fax. (301) 353-0731